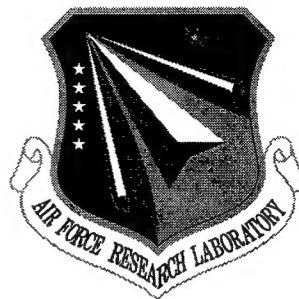


AFRL-SN-RS-TR-1999-96
Final Technical Report
May 1999



APPLICATION ANALYSIS FOR ELECTRO-OPTIC INTERCONNECT COUPLING

Defense Group, Inc.

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. B667

19990629 087

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
SENSORS DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

DTIC QUALITY INSPECTED 4

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-SN-RS-TR-1999-96 has been reviewed and is approved for publication.

APPROVED:



KEVIN J. WHITCOMB, Capt, USAF
Project Engineer

FOR THE DIRECTOR:



ROBERT G. POLCE, Acting Chief
Rome Operations Office
Sensors Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/SNDP, 25 Electronic Parkway, Rome, NY 13441-4515. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

APPLICATION ANALYSIS FOR ELECTRO-OPTIC INTERCONNECT COUPLING

Morgan Grover
Pankaj Patel

Contractor: Defense Group, Inc.
Contract Number: F30602-97-C-0295
Effective Date of Contract: 12 August 1997
Contract Expiration Date: 12 April 1998
Short Title of Work: Application Analysis for Electro-Optic
Interconnect Coupling
Period of Work Covered: Aug 97 - Apr 98

Principal Investigator: Morgan Grover
Phone: (310) 394-8599
AFRL Project Engineer: Capt. Kevin J. Whitcomb
Phone: (315) 330-1821

Approved for public release; distribution unlimited.

This research was supported by the Defense Advanced Research
Projects Agency of the Department of Defense and was monitored by
Capt. Kevin J. Whitcomb, AFRL/SNDP, 25 Electronic Parkway, Rome,
NY 13441-4515.

Table of Contents

List of Figures	ii
List of Tables.....	iii
1. INTRODUCTION AND OVERVIEW	1
2. SAR IMAGE FORMATION PROCESSING STUDIES.....	4
2.1. BASIC STRIPMAP OVERVIEW	5
2.2. SAR CASES FOR PRESENT STUDY	16
2.3. CONSIDERATIONS FOR SPOTLIGHT IMAGING	23
2.4. PROCESSING HARDWARE ASSUMPTIONS	31
2.5. SUMMARY OF ANALYSIS RESULTS.....	32
3. AUTOMATIC TARGET RECOGNITION (ATR).....	39
3.1 INITIAL STUDY.....	40
3.2 DETAILED STUDY	45
3.2.1 <i>Overview of Basic Approach and Summary of Prior Results:</i>	46
3.2.2 <i>Technical Description of the Approach:</i>	51
3.2.3 <i>Computing Power and Memory Requirement:</i>	56
3.2.4 <i>Real time Implementation:</i>	63
4. DISCUSSION	68

List of Figures

Figure 1.1. Top-level motivation of DARPA E/O interconnect and switching technology development programs.....	1
Figure 2.1. Radar-platform/target-scene viewing geometry: a) 3-D view; b) side view. ..	6
Figure 2.2. Radar-platform/target-scene viewing geometry: top view.	8
Figure 2.3. Range-curvature representation.	9
Figure 2.4. Basic stripmap SAR processing schematic.	12
Figure 2.5. High level schematic including autofocus and multi-look.	13
Figure 2.6. Schematic of Phase Gradient Autofocus Steps.....	15
Figure 2.7. Schematic of five subaperture positions relative to input target.....	16
Figure 2.8. I/Q baseband demodulation.	23
Figure 2.9. Basic comparison of spotlight and strip map SAR imaging.....	24
Figure 2.10. Main processing stages for Polar Format Algorithm, Range Migration Algorithm, and Chirp Scaling Algorithm for spotlight SAR imaging.....	25
Figure 2.11. E/O crossbar switch for SAR image formation processing.	38
Figure 3.1 A basic structure for ATR processing.	42
Figure 3.2 Comparison of computing power required by prior, current and future ATR methods.....	43
Figure 3.3 Prior, current and future computing versus bandwidth.	44
Figure 3.4 Prior, current and projected future computing versus bandwidth trades for automatic target recognition.....	45
Figure 3.5 Most basic multi-stage ATR processing chain, showing both on-line and off-line components.....	47
Figure 3.6 First stage FAR/PFR versus detection metrics based on MSTAR data.	47
Figure 3.7 PFA/PFR for initial target detection. FLIR at left, MSTAR at right.....	48
Figure 3.8 Classification/indexing results at initial target detection stage. FLIR results at left, SAR results at right.....	49
Figure 3.9 Final stage PFA/PFR results for FLIR data on left and SAR on right.	50
Figure 3.10 Final classification results for FLIR data on left and SAR data on right.	50
Figure 3.11 Initial processing for scene-adaptive clutter identification and modeling....	52
Figure 3.12 Sample distributed architecture for implementing the end-to-end ATR.	64

List of Tables

Table 2.1. Nominal baseline parameters for Tier II and Tier III radars.....	17
Table 2.2. Initial summary of SAR parameters for cases of present study.....	19
Table 2.3. Completed summary of SAR parameters for cases of present study.	20
Table 2.4. Impacts of using range dechirp on size/number of required swaths.	22
Table 2.5. Limitations on PFA scene size for cases in present study.	26
Table 2.6. Scene sizes for appreciable range skew for the PFA and RMA.	28
Table 2.7. Operations per output pixel for three different spotlight SAR imaging modes, for specific conditions given in the text.....	30
Table 2.8. Initial summary of basic range compression processing parameters.	33
Table 2.9. Summary of basic processing factors for azimuth compression.....	35
Table 3.1. Prior, current and future pixel rate and input resolution.....	56
Table 3.2. Computing and memory for anomaly detector stage versus pixel rate.	58
Table 3.3. Computing and memory associated with preliminary target detector stage versus pixel rate.....	59
Table 3.4. Computing and memory associated with target/clutter classifier stage versus pixel rate.	60
Table 3.5. Computing and memory for target/target stage versus pixel rate.	61
Table 3.6. Summary of computation power and memory for a pixel rate of 2×10^7	62
Table 3.7. Summary of computation power and memory for a pixel rate of 2×10^8	62
Table 3.8. Summary of computation power and memory for a pixel rate of 2×10^6	63

1. INTRODUCTION AND OVERVIEW

This report documents an exploratory study of potential impacts of electro-optic (E/O) interconnect and switching technologies for highly demanding defense computing applications. The analyses reported below address both future high-resolution wide-area synthetic aperture radar (SAR) image formation processing, and also processing for advanced methods for automatic target recognition (ATR) at high input pixel rates. Similar analyses for moving target detection and tracking have also been performed, but are not reported.

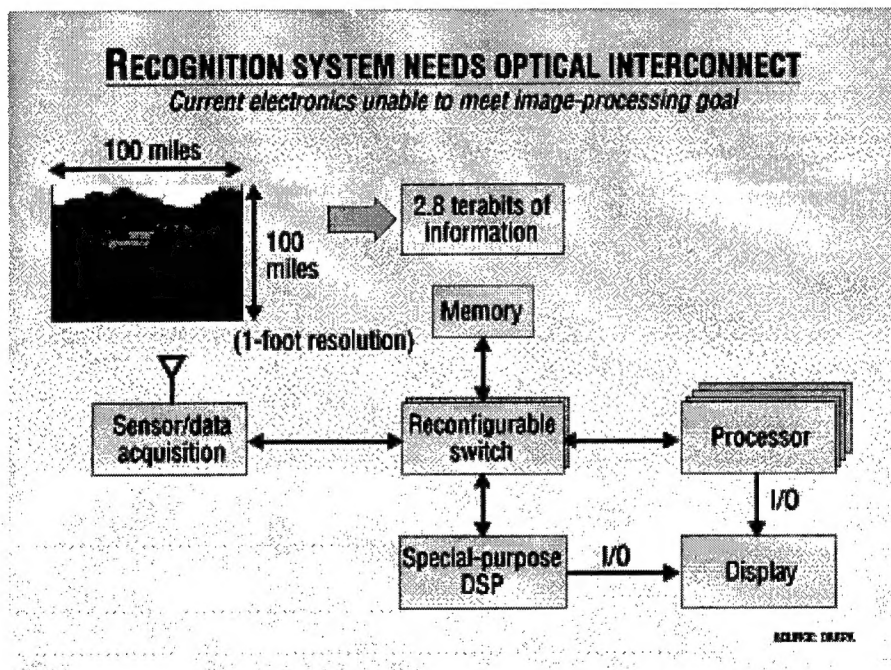


Figure 1.1. Top-level motivation of DARPA E/O interconnect and switching technology development programs.

The motivations for this study came from the DARPA Free Space Optical Interconnect program, and its follow-on VLSI Photonics program. The goals of these programs are to develop E/O interconnect switching and related technologies to accomplish future signal processing applications such as those addressed here. While very "top-level" motivations for these programs have previously been

available, as specifically exemplified by the above figure, most of the technical details have been lacking. The DARPA program managers therefore recognized a need for more detailed analyses of how E/O technologies might actually enhance specific processing applications, both for purposes of "program defense" and also for focussing and directing the E/O technology development. The present effort was contracted to help fill this gap.

The resources provided for this effort were very limited (\$39K), and a noteworthy portion was ultimately required to support DARPA program meetings. The initial planned scope of work under this funding was limited to a preliminary investigation of E/O technology impacts for SAR processing. Both ATR and tracking were initially considered as options for added funding.

From the outset, however, we were able to provide the then-current DARPA program manager (Dr. A. Hussain) with results and briefing materials from initial analyses of the potential benefits of E/O interconnect and switching for both SAR and ATR applications, and we received feedback that this breadth of scope had been highly valuable for his defense of both the Free Space Optical Interconnect and follow-on VLSI Photonics programs. This feedback was repeated at the initial DARPA program review. At that review we received encouragement to carry on with a broad scope of investigation in anticipation of additional funding and an extended period of contract performance.

In the end, neither the added funding or extended period of performance were ever realized. In the meantime, however, we have proceeded as well as possible with our exploratory studies of the potential benefits of E/O interconnect technologies for: (1) SAR image formation for high-resolution wide-area search, (2) advanced methods for ATR at high pixel input rates, and (3) moving target detection and tracking at high input pixel rates, including advanced methods for "track before

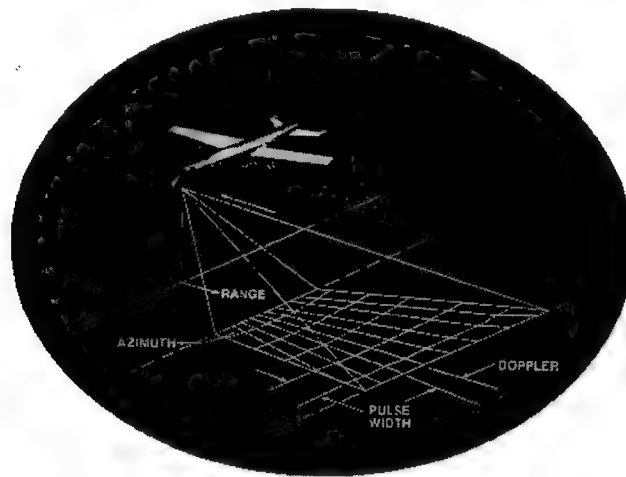
detect" processing for improved clutter and false alarm rejection. Our preliminary findings in the former two application areas are reported here.

For SAR imaging we have been able to identify how processing architectures built around the envisioned E/O crossbar switching has potential to allow for real-time SAR imaging at high resolutions and for wide area search with a very small overhead in terms of the amount of computing power required. For ATR we have found that future processing architectures incorporating envisioned E/O crossbar switching technologies can potentially allow for very efficient utilization of the available processing power, and can also significantly reduce the total memory requirements -- which are another major factor for high performance ATR applications. Similar potential benefits from envisioned E/O crossbar switching technologies have been identified for moving target detection and tracking with track before detect filtering, but are not reported here.

Section 2 of this report reviews our studies of SAR image formation processing, and of the potential use and impacts of E/O crossbar switching to allow this processing to be done with very small overhead in terms of the required computing power for its execution. Section 3 of this report provides a similar documentation of our studies of advanced methods for ATR processing at high input pixel rates, and describes conceptual processing architectures based in part on high-speed E/O crossbar switching which have potential to allow both a very high level of processor utilization and also significant economies in terms of the required amount of solid-state random access memory for this class of applications. A summary and discussion of the overall findings from this study is given in Section 4.

2. SAR IMAGE FORMATION PROCESSING STUDIES

In this section we describe our initial studies of synthetic aperture radar (SAR) image formation processing for future, high resolution and wide area search SAR systems, and the utility of E/O interconnect technologies for real-time execution of the required processing. Very early results from this work were provided to the DARPA program manager, and were successfully used to promote and defend both the Free Space Optical Interconnect and VLSI Photonics programs.



Subsection 2.1 provides an overview of basic stripmap processing for wide-area search. Subsection 2.2 describes the specific cases chosen for present study and their relationship to the developmental Tier II and Tier III systems, and to other existing experimental systems for high resolution SAR imaging (typically also with limited wide-area search capabilities). Subsection 2.3 discusses some of the pros and cons of stripmap versus spotlight imaging methods for high-resolution, wide-area search applications. Subsection 2.4 addresses hardware, and then Subsection 2.5 addresses numerous implementation details and processing and communications burdens for the envisioned future SAR applications, and a general processing architecture by which this can potentially be done with very low computing overhead based on the use of one or more high-speed, non-blocking E/O interconnect crossbar switches. A summary discussion is also given in Section 4.

2.1. Basic Stripmap Overview

The basic stripmap SAR algorithm is described in this subsection. “Strip-map-mode” refers to the SAR operational mode in which a strip of ground is imaged by a radar system moving overhead, such as on an aircraft or spacecraft, with fixed heading and with its radar-beam orientation fixed relative to the platform. The imaged ground area is swept over by the radar beam of the moving platform as consecutive radar pulses are emitted and their echoes received.

Figure 2.1 shows the radar-platform/target-scene viewing geometry. The radar-platform is assumed to be moving parallel to the earth’s surface at a constant speed, heading, and altitude. The earth’s surface is assumed to be flat and non-rotating. The radar-beam angle is 90° with respect to the flight path, and is at a specified angle θ relative to nadir. The projection of the radar-platform’s flight path onto the ground defines the azimuth or “cross-range” dimension. The range dimension is defined to be perpendicular to the azimuth dimension. The distance from platform to ground along any line is known as “slant range.”

As the radar-platform moves along its flight path, many radar pulses are transmitted and received, typically at a rate greater than 300 Hz. In SAR image formation, many received pulses are processed together in a way that produces an image of the radar-illuminated area with a much higher resolution than is possible using a single pulse. After compensation for the delay difference between each received pulse and a reference pulse, the received pulses can be coherently added to form an image. The distance traveled by the radar-platform during acquisition of the pulse data used in the coherent integration determines the length of the synthetic aperture of the radar.

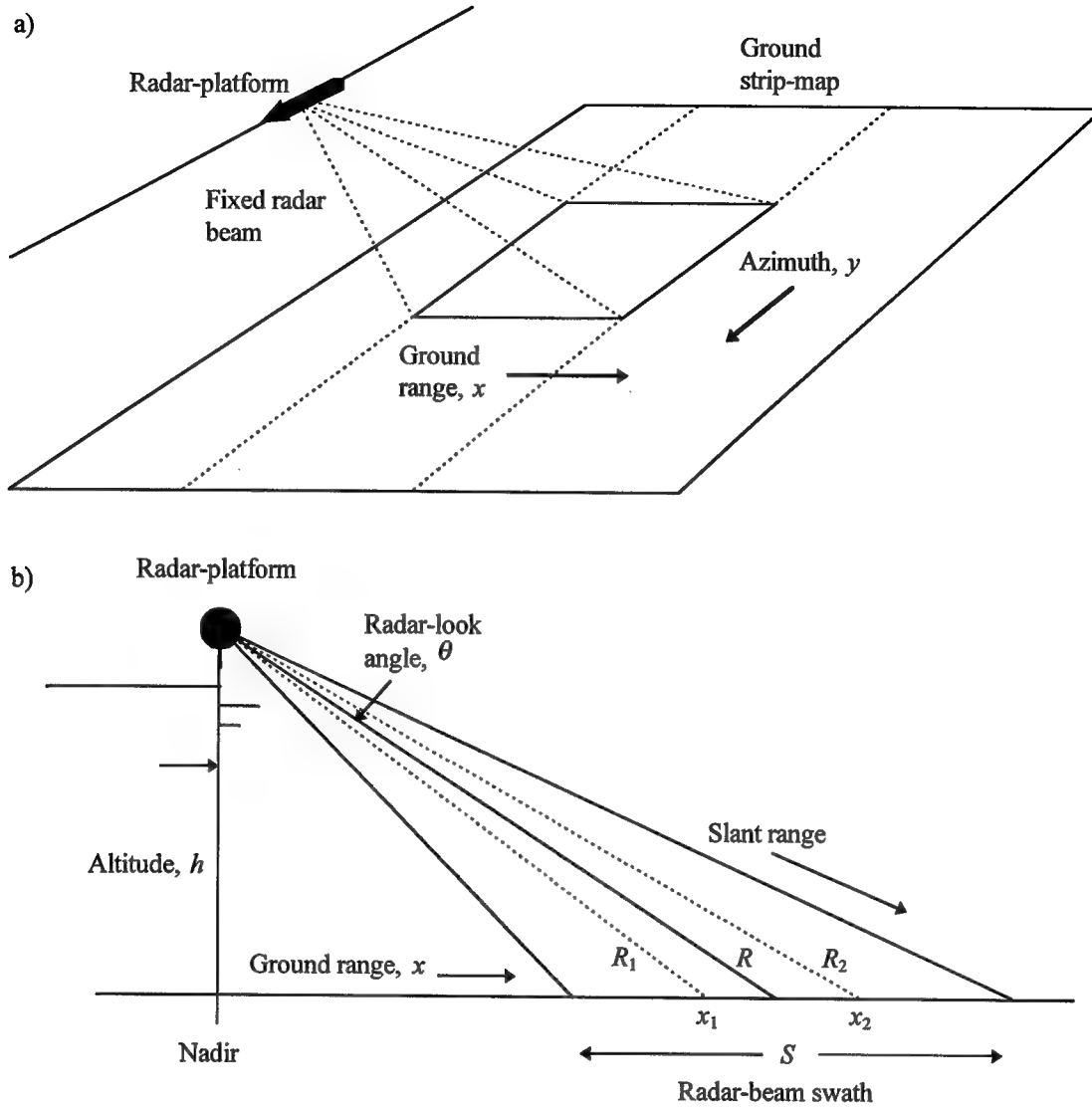


Figure 2.1. Radar-platform/target-scene viewing geometry: a) 3-D view; b) side view.

The transmitted pulse is taken to be a frequency-modulated sinusoidal signal, also known as a “chirp.” In its amplitude-normalized form, it is the real part of

$$s(t) = \begin{cases} \exp[j(\omega_0 t + \pi \gamma t^2)], & |t| \leq \frac{T}{2} \\ 0, & \text{otherwise} \end{cases}, \quad (2.1)$$

where t is time; T is the time-width of the pulse; ω_0 is the angular carrier-frequency ($=2\pi f_0$, where f_0 is the radar carrier-frequency); and γ is the chirp rate, where

$$\gamma \approx B/T, \quad (2.2)$$

with B defined as the frequency bandwidth of the chirp.

The return signal from a single pulse is the sum of all the returns from scatterers illuminated by the beam:

$$r(t) = \iint dx dy \frac{G(x, y)}{[r(x, y)]^2} g(x, y) s \left(t - \frac{2r(x, y)}{c} \right), \quad (2.3)$$

where $r(x, y)$ is the distance between radar-platform and a scatterer located on the ground at (x, y) ; $G(x, y)$ is the two-way antenna pattern (power gain); c is the speed of light; and $g(x, y)$ is a complex function whose magnitude is the fraction of incident radiation reflected back to the radar and whose phase is due to the shift that can occur when the radar wave is reflected, due to material properties and varying elevations of the air/target interface. The goal of SAR-processing is to reconstruct $g(x, y)$, and it is the magnitude of this function that is displayed as the SAR image.

The time interval over which the return signal is acquired determines the range coverage of the SAR data processing. In order to generate a SAR image for the area depicted in Figure 2.2, the return signal must be acquired in the time interval

$$-\frac{T}{2} + \frac{2R_1}{c} \leq t \leq \frac{T}{2} + \frac{2R_2}{c}, \quad (2.4)$$

where slant ranges R_1 and R_2 are the shortest and longest distances between radar-platform positions within the synthetic aperture and scatterers in the imaged area.

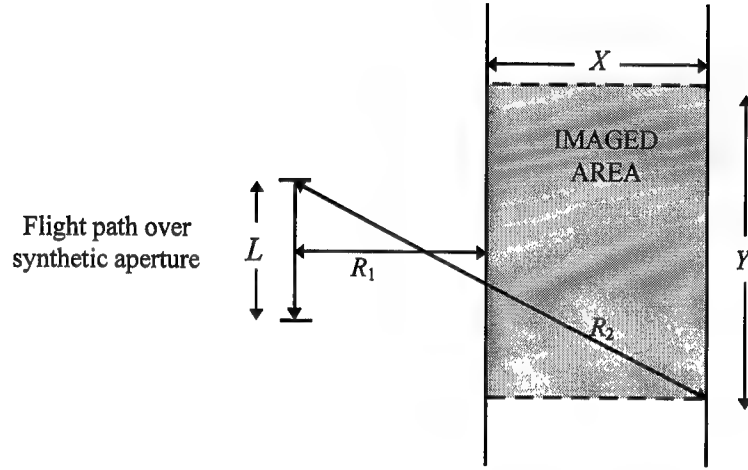


Figure 2.2. Radar-platform/target-scene viewing geometry: top view.

By correlating each return signal, with the corresponding transmitted signal at a delay of $t_n = 2r_n/c$, the total signal strength from scatterers at slant range r_n is obtained. This type of operation, known as range compression, transforms the return signal of time-width T into a compressed pulse that has a sinc-type behavior in time. The half-width of this function in time as measured from the pulse center to the first null is $\tau = 1/B$. The amount of pulse compression obtained is determined by the bandwidth of the transmitted signal. The more the received pulse can be compressed by using larger values of B , the better the range resolution. The corresponding “slant-range” resolution, Δr , and “ground-range” resolution, Δr_g , are given by

$$\Delta r = \frac{c\tau}{2} = \frac{c}{2B},$$

and

(2.5)

$$\Delta r_g = \frac{\Delta r}{\sin \theta},$$

where θ is the angle between the center of the beam and nadir.

Azimuth compression, the next step in stripmap SAR image formation, is the process of forming an image of a ground point by coherently adding pulse-compressed radar samples from multiple consecutive pulses, which have each been gated at the slant-range delay corresponding to the separation of the radar platform and ground point. If these delays differ from one another by more than the spacing between range samples, then what is known as “range curvature,” as illustrated in Figure 2.3, affects how the azimuth compression is done.

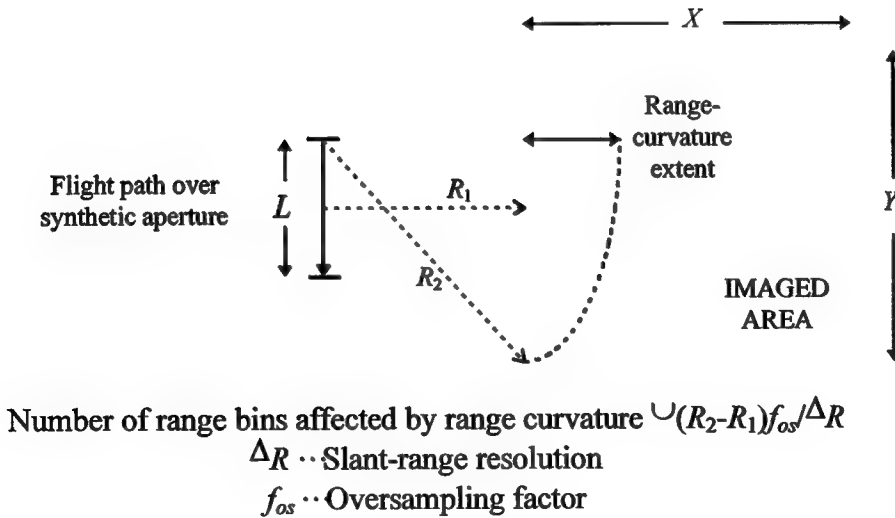


Figure 2.3. Range-curvature representation.

Compensation for the phase difference between a sample for a given delay, and for a given azimuth position, versus a sample from the reference pulse, is made by multiplying each (complex) sample by the corresponding complex exponential containing the negative of the corresponding phase difference. This multiplier is called the azimuth reference function. Azimuth compression is mathematically equivalent to correlating pulse-compressed samples at the appropriate delays with the azimuth reference function for those delays. The magnitudes of the complex output of the azimuth compression algorithm form a SAR image.

The azimuth resolution of the resulting SAR image depends on a number of system parameters that figure into the azimuth compression process. Assuming that the chosen azimuth resolution is Δr_a , the corresponding synthetic-aperture length, L , is on the order of

$$L \approx vT_D = \frac{\lambda R}{2\Delta r_a}, \quad (2.6)$$

where λ is the radar wavelength, equal to c/f_0 ; R is the side-looking slant range to the azimuth line being imaged (see Figure 2.1.b); v is the radar-platform speed relative to the stationary ground; and T_D is the “coherent integration time,” i.e., the flight-time of the radar-platform across the synthetic aperture. Since an imaged ground point must be illuminated by the beam over the entire coherent integration period, L can be no longer than the azimuth length of the beam on the ground, which is on the order of $\lambda R/l$, where l is the azimuth length of the antenna. Therefore, the best possible azimuth resolution is on the order of $\Delta r_a \geq l/2$.

The pulse repetition frequency, PRF , is a system parameter whose value must be chosen sufficiently large to avoid the problem of azimuth “ambiguities.” These are multiple ghost images of azimuthally-offset ground areas which overlap the

SAR image of the desired area. These unwanted artifacts arise due to the finite number of discrete samples from different pulses that are coherently added during azimuth compression. This number is equal the number, N_p , of pulses transmitted during radar-platform flight across a synthetic aperture which is the product of the flight-time across the synthetic aperture and the PRF . The azimuth offset distance on the ground between an ambiguous area and the area to be imaged is called the ambiguity spacing. It can be shown that the ambiguity spacing, Δy , is directly proportional to N_p , and therefore to PRF :

$$\Delta y = \frac{\lambda R}{2L} N_p = \Delta r_a N_p = \Delta r_a T_D PRF. \quad (2.7)$$

The PRF must be chosen large enough that Δy is larger than the projection of the radar beam on the ground in the azimuth dimension. The resulting lower bound on the PRF can be expressed as

$$PRF \gtrsim \frac{2v}{l}. \quad (2.8)$$

The PRF is normally also constrained by the requirement that all returns in a single pulse from scatterers in the beam must be received in the interval between two successive pulse transmissions. If the closest and farthest scatterers in the beam relative to the radar-platform are at slant ranges R_{\min} and R_{\max} , then

$$PRF \lesssim \frac{c}{2(R_{\max} - R_{\min})}. \quad (2.9)$$

Both pulse and azimuth data-compression operations are mathematically equivalent to correlations, which are efficiently handled using fast Fourier transform (FFT) methods. The computational scheme for generating single-look SAR images is depicted in Figure 2.4. It consists of pulse compression, a range-to-azimuth “corner-turn,” and azimuth compression

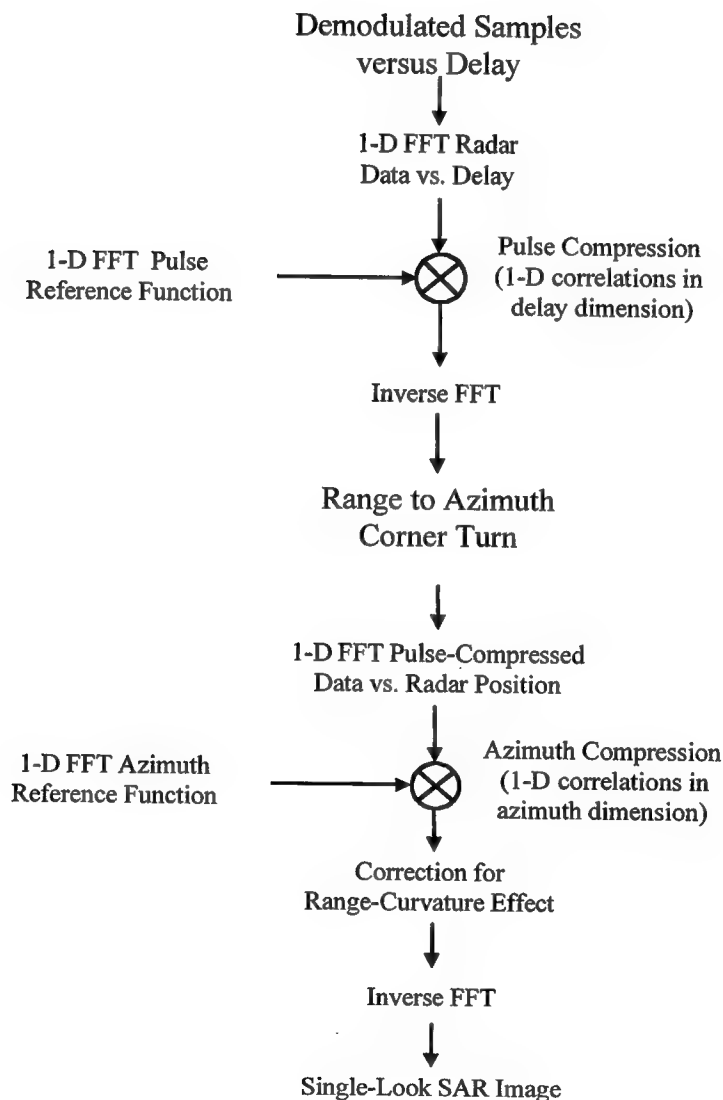


Figure 2.4. Basic stripmap SAR processing schematic.

Two further considerations for the implementation of a high resolution SAR are autofocus and multi-look averaging. Autofocus is considered essential for high resolution systems. Conversely, multi-look averaging for speckle suppression may often be unaffordable. These points are briefly discussed below.

Achieving and maintaining very high SAR spatial resolution is typically not possible based only on platform guidance feedback to the receiver from the on-board inertial navigation unit(s) and GPS. For this reason it is necessary to actually evaluate the SAR image to obtain an estimate of the residual signal phase errors, and to use this estimate to improve the focussing. This typically must be done in several iterations. Figure 2.5 gives a high level schematic of the process, and also shows potential use of multi-look averaging for the final product image.

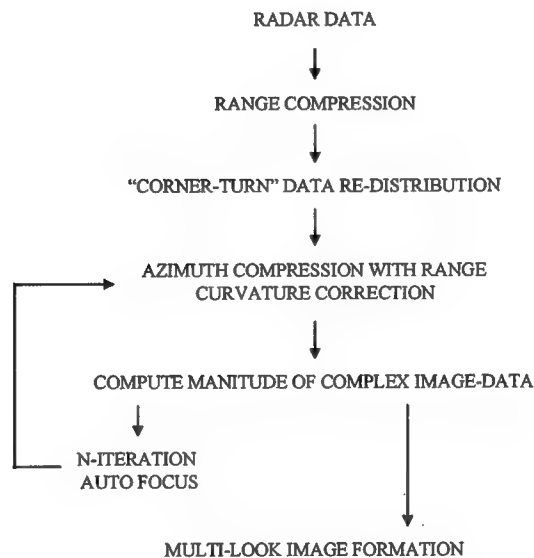


Figure 2.5. High level schematic including autofocus and multi-look.

Two general approaches to autofocus can be considered. The first is based on forming a series of images of the same scene with displaced synthetic aperture centers, and correlating the resulting images to find their relative displacements as imaged from one aperture to the next. This then yields a series of estimates of the

residual quadratic phase error, at the midpoint between each pair of apertures, which can be integrated versus azimuth radar location and then used to refine the next iteration of azimuth focussing. This method is relatively expensive, and is most commonly used with only two apertures and for detecting and correcting an overall quadratic phase error, although higher orders of phase error can also be corrected at the expense of using more displaced apertures.

A more commonly used method for detecting and correcting higher order residual phase errors is the Phase Gradient Algorithm, which is depicted in Figure 2.6. It is based on the assumption that the imaged region can be thought of as a collection of point targets. In the image domain the brightest point on each range line is selected and then windowed versus azimuth. The complex windowed range line image is then rotated so that the bright target is in the center of the image, and the final azimuth Fourier transform is inverted. Then, an averaging process over all range lines (or a sufficient number) thus processed is used to derive an estimate of either a first or second derivative of the overall phase error versus radar location along the synthetic aperture. This estimate can then be used to refine the azimuth focusing in a second iteration. The cost of the algorithm (aside from the final refocussing step) can be kept small by the windowing, since the sizes of the inverse transforms can also be reduced accordingly, and progressively further reduced on each subsequent iteration. A summary of the algorithm is given in Figure 2.6.

Speckle noise, which gives SAR imagery a grainy appearance, is due to the coherent nature of the SAR image formation process. Multi-look SAR processing is a commonly employed technique for reducing speckle noise, and involves the incoherent addition of two or more “looks” at the same target scene, obtained from processing different, independent sets of radar returns. However, when the looks are generated by partitioning the synthetic aperture into subapertures, as illustrated

in Figure 27, the shorter subapertures cause a degradation in the azimuth resolution. The azimuth-resolution degradation realized by generating a given number of looks must be traded off with the speckle-noise reduction achieved by incoherently adding that number of looks.

For that reason, we will not address multi-look averaging in the present study, but note that for those candidate systems which also have multiple radar polarizations an incoherent adding of images formed for each received signal polarization can also be used as a means of speckle reduction.

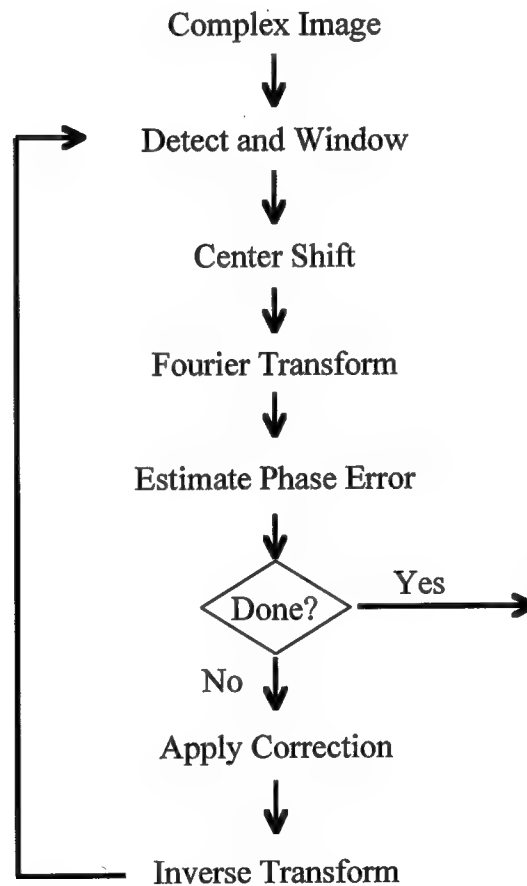


Figure 2.6. Schematic of Phase Gradient Autofocus Steps

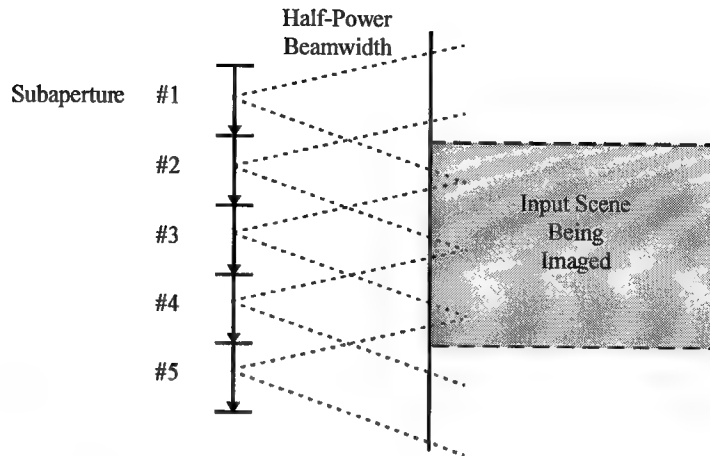


Figure 2.7. Schematic of five subaperture positions relative to input target

2.2. SAR Cases for present study

In examining the future utility of E/O switching technology for future SAR image formation processing applications, the anticipated timelines for the development of this technology require that the applications addressed be projected, future, high-performance SAR applications, including cases that are potentially well beyond the capabilities of current technology.

The present study has addressed several examples of SAR imaging for wide area search at very high resolution. For the most part we have chosen not to focus directly on specific existing experimental or near-term developmental systems, but to use these systems as a point of departure, mainly because such systems are already within (or close to) the realm of feasibility based on current technology; although major limitations do still exist in the sizes of the range swaths that can be dealt with, and thus the total area search rates that can be provided.

In the present study we have chosen mainly to address potential future systems with higher spatial resolution, larger range swaths, and generally also a larger number of polarization channels than current and developmental systems such as

the developmental Tier II and Tier III radar systems, or the experimental ADTS and Twin-Otter systems of MIT Lincoln Laboratory and Sandia, respectively. We have, however, taken guidance from the general properties of these existing systems, as briefly summarized for both Tier II and Tier III in the following table. Note that some of the following parameters are based on discussions with Northrop and Hughes, some are derived from the literature, and some are also our own estimates or inferences.

System	Tier II	Tier III
Altitude (km)	<20	<17
Frequency (GHz)	9.6	16.5
Wavelength (cm)	3.1	1.8
Nominal Speed (m/s)	178	130
Antenna Length (cm)	122	91
Antenna Width (cm)	37	27
Grazing Angle (°)	7-30	10-30
Strip Resolution (cm)	61	46
Oversampling (%)	20	20
Polarizations	1	1
Bandwidth (MHz)	300	326

Table 2.1. Nominal baseline parameters for Tier II and Tier III radars.

In consideration of the range of grazing angles indicated above, it is clear that the total range swath illuminated by the radar will be greatest for the smallest grazing angles, and that this situation would theoretically allow the highest search rates, provided that the entire illuminated swath can actually be imaged. In practice, however, the systems summarized above are currently limited in the sizes and

numbers of the range swaths that can be processed, and improvements in this regard are one of the main objectives of future technology development. The same is true for the experimental Lincoln ADTS system, which reportedly has a range swath size of only 375 m for stripmap imaging with 1 ft. or poorer resolution, and for the Sandia Twin Otter system, with a reported maximum range swath of 1792 pixels or less at 1 ft. or poorer resolution. Thus, in addition to improving sensor resolution (also the number of polarization channels supported) a major need for future SAR imaging technology is also to increase the width and/or number of range swaths that can be imaged.

With the above as background, we have selected the following three cases for the present study. These involve SAR resolutions equal to or better than the current developmental and experimental systems discussed above, also much wider range swaths, and also an increasing number of signal polarizations to be processed. To maximize the range swath we are considering here only a rather small grazing angle of 10° , and imaging of the entire illuminated swath. As a reasonable compromise, consistent with the differences between Tier II and Tier III in this regard, we are also envisioning a trend to lower operating altitudes, and higher operating frequencies, as the basic SAR resolution improves.

Most of the entries in Table 2.2 are self-explanatory. One point, which relates also to Table 2.1, is that the listed platform speed is a nominal value. We assume that the radar PRF, the minimum required value of which varies with platform speed, will either be adjusted by the on-board INS or else oversampled and then resampled prior to the SAR image formation. The latter approach involves noteworthy 2D data rearrangements in the receiver front-end, but nothing comparable to the 2D data rearrangements needed in the actual image formation processing; so, we will not address it further in this study.

Case #	1	2	3
Altitude (km)	20	15	10
Nominal Speed (m/s)	170	170	170
Grazing Angle (°)	10	10	10
Polarizations	1	2	3
Oversampling (%)	20	20	20
Slant Range (km)	115	86.4	57.6
Ground Range (km)	113	85.1	56.7
Frequency (GHz)	10	20	40
Wavelength (cm)	3	1.5	.75
Ant. Length (cm)	122	61	30.5
Ant. Width (cm)	40	30	20
Az. Beamwidth (°)	1.41	1.41	1.41
El. Beamwidth (°)	4.3	2.87	2.15

Table 2.2. Initial summary of SAR parameters for cases of present study.

Concerning Table 2.3, note that the “Average Azimuth Width” is the along-track width of the illuminated scene at the middle of the range swath. It is also the synthetic aperture length for best azimuth resolution at that point in range. The illuminated scene width at closer and farther ranges, and also the best synthetic aperture length, will vary (by ± 11 -23%) for the different cases addressed. This difference will not have a major effect on the SAR azimuth processing, however, since the synthetic apertures will be zero-padded up to the next higher power of 2 to allow use of fast Fourier transform methods.

Case #	1	2	3
Az. Resolution (cm)	61	30.5	15.2
Avg. Az. Width (km)	2.83	2.12	1.42
Az. Overlap (%)	50	50	50
Avg. Pix./Az. x O/S.	5567	8341	1182
Slant Res. (cm)	61	30.5	15.2
Slant Swath (km)	51.4	25.1	12.4
Slant Pix./Pulse x O/S	1.01E5	9.88E4	9.79E4
B (MHz) x O/S	295	590	1184
PRF (Hz)	334	669	1114
T_p (μ s)	50	30	20
Chirp Rate (MHz/ μ s)	4.92	16.4	49.3
# Range Curvature	14	21	29

Table 2.3. Completed summary of SAR parameters for cases of present study.

Also, since only half of this scene width is strongly illuminated by the radar as it moves over the full synthetic aperture, we are assuming a factor of 1.5 overlap between successive apertures. This will lead to a two-times increase in the azimuth focussing relative to no overlap, but it has no effect on the range focussing, which needs to be done only once for every transmitted/received pulse.

Notice that there is appreciable range curvature in the cases listed above. This will have a significant impact on the azimuth focussing, since an interpolation over the range of ranges seen at a given delay will be required. The range curvature values listed above in terms of numbers of range cells do not include range oversampling, since we envision that this oversampling will be undone after the range focussing,

and also do not include curvature for points outside the main imaged area of each overlapping synthetic aperture.

The assumed PRFs in each case are close to the minimum acceptable for prevention of azimuth ambiguities, but do have a 20% margin due to the azimuth oversampling. The number of range pixels per pulse is very large. Given the finite pulse lengths assumed, the range compression Fourier transforms can most optimally be done as a succession of transforms for different range-gated parts of the entire return. In the following, however, we will assume that this has not been done, since its optimization depends on the specific transmitted pulse width, and the values listed above for this parameter are only initial estimates – and may be too small.

A final major consideration from the above is the bandwidth(s) required for the assumed range (and azimuth) resolutions. The values listed include a 20% margin for range oversampling. For the basic stripmap approach being considered here this would require an A-to-D sampling rate equal to the listed bandwidth(s). An alternate approach would be to use range dechirp receiver processing, in which signals from different range subswaths are separately demodulated as different channels, and the required sampling rate per channel can then be reduced by a factor of

$$B_{IF}/B \approx 2\Delta R/cT_p \quad (2.10)$$

Here, B_{IF} is the receiver intermediate frequency bandwidth for each channel from the dechirp demodulation, and ΔR is the width of the range subswath addressed.

This approach is used in most current developmental and experimental high resolution SAR systems. One of its problems is a phenomenon called “residual

video phase,” which will be discussed in the next subsection. Another problem is the limitation it puts on the size of each individual (also overlapping) range swath, and thus also the number of separate receiver channels that would be required to cover the entire illuminated range swath of the cases listed above. This point is addressed in the following table for the cases listed above, and on the assumption of a 150 MHz intermediate frequency, just slightly larger than the 125 Msps sampling rate of the Lincoln ADTS experimental system.

Case #	1	2	3
Subswath (km)	4.57	1.37	0.456
Number Required	12	19	28

Table 2.4. Impacts of using range dechirp on size/number of required swaths.

As illustrated in Figure 2.8, the alternate to the above range dechirp receiver processing, and its limitations on individual range swath width, and requirement for multiple receiver channels for each range subswath, is to directly convert the received signal down to baseband, and sample the resulting in-phase and quadrature (I and Q) channels at a rate equal to the full bandwidth plus the range oversampling.

The difficulty here has been the lack of sufficiently fast ADCs. This, however, is changing; and the needed ADC support for the SAR cases listed above is expected to be fully available on a time scale consistent with the practical implementation of the E/O interconnect technologies being addressed in this study. As already noted, the Lincoln ADTS system, which has been flying for over a decade, already uses an A/D rate of 125 Msps. Currently, Analog Devices sells their 200 Msps (380 MHz bandwidth) AD9054 ADC chip. Harris also offers their 500 Msps HI1276 ADC chip. And where it is really needed, for very high speed digital

oscilloscopes for radar and related applications, both LeCroy and Tectronix currently offer single-channel and multi-channel sampling rates of 1-5 Gsps using ADCs operating at 1-1.24 GSPS. For these reasons, and in view of the anticipated time scales for development of the E/O interconnect technology, and other considerations regarding range-chip demodulation to be discussed in the next section, we have elected in the present study to anticipate the availability of the required ADC technology consistent with the E/O technology being addressed.

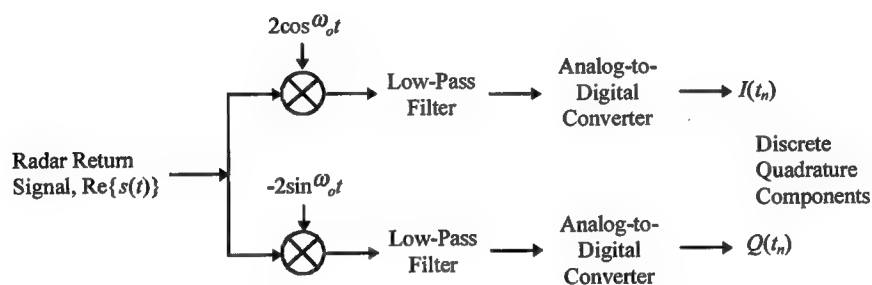


Figure 2.8. I/Q baseband demodulation.

2.3. Considerations FOR Spotlight Imaging

The differences between stripmap and spotlight SAR imaging in general are variable and negotiable depending on the specific implementation. As indicated in Figure 2.9, basic spotlight imaging nominally assumes a radar beam that is continually pointed at the center of the scene being imaged, while stripmap imaging allows the beam center to move with the radar platform. In fact, the differences between these two perspectives for wide area search are only relevant when the spotlight aperture length is much longer than the stripmap aperture length over which the same scene is illuminated by the radar, which mainly happens

when the signal bandwidth allows a cross-track resolution greater than can be achieved in the along track dimension by a conventional stripmap imaging mode. This condition also allows for a lower PRF for the spotlight mode, since the time over which multiple returns are accumulated is extended. However, for the cases of present interest there is generally no excess in available signal bandwidth; and the aperture lengths for both spotlight and stripmap imaging are essentially the same.

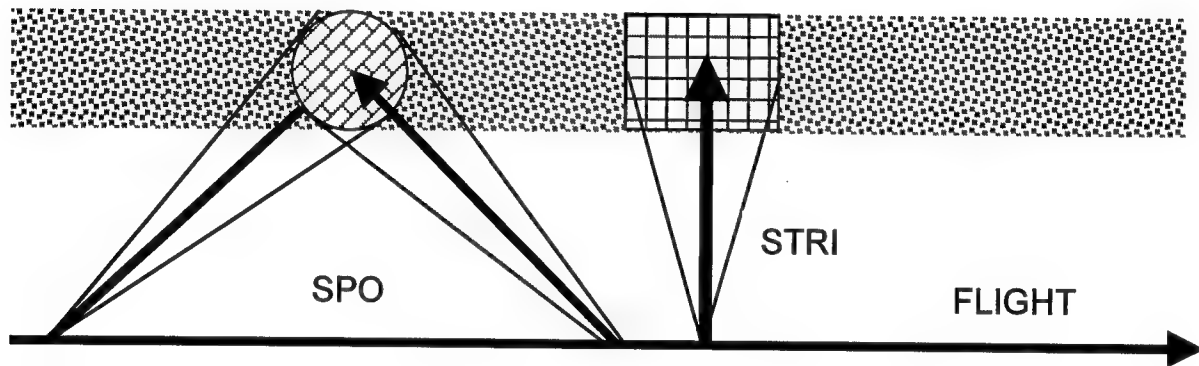


Figure 2.9. Basic comparison of spotlight and strip map SAR imaging.

Another general difference between spotlight versus stripmap SAR imaging is in how the received radar signals are demodulated. This, however, also depends on how the subsequent SAR image processing is intended to be performed. In the most basic stripmap mode the received signals are simply down-converted and I/Q sampled at the full rate of the transmitted signal. In the spotlight Polar Formatting Algorithm approach (PFA -- discussed below) the received signals are dechirped in both range and azimuth by mixing with a reference function for this purpose, and then A/D sampled at a lower rate. A similar dechirp demodulation is used for the Range Migration algorithm (RMA -- below); while for the Chirp Scaling Algorithm (CSA) the nominal demodulation is the same as for basic stripmap imaging.

Figure 2.10 provides an overall summary of the major processing steps in a variety of different spotlight SAR processing modes. The additional data movement and corner turn requirements are further discussed below.

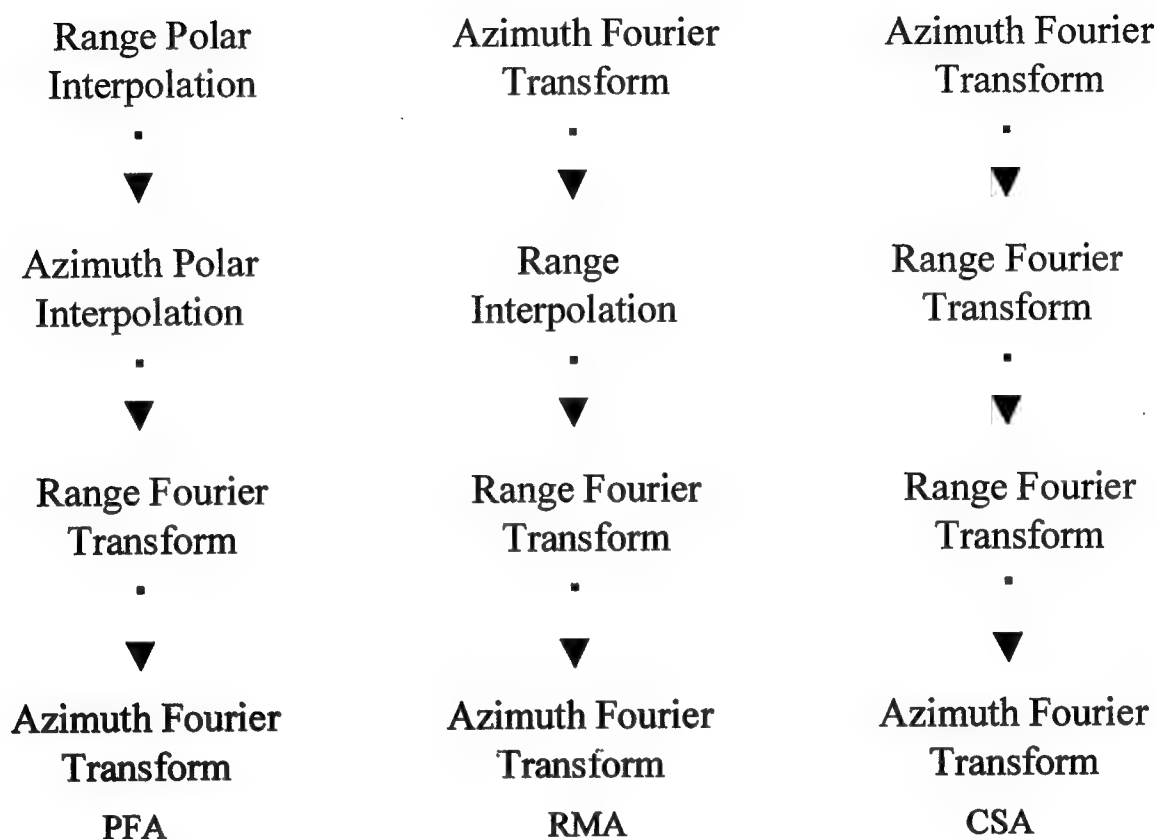


Figure 2.10. Main processing stages for Polar Format Algorithm, Range Migration Algorithm, and Chirp Scaling Algorithm for spotlight SAR imaging.

The PFA uses radar signals after removal of the chirp in both range and azimuth, which provides compensation relative to a single point at the center of the scene or subscene being imaged, converting the signal from the time to frequency domain in both range and azimuth, and ideally permitting the imaging to be completed by a 2D Fourier transform. Except for very small scenes, however, PFA also requires 2D polar format interpolation in both range and azimuth to partially compensate for curvature effects. For the range of (small) viewing angles normally used, this

interpolation can be done as a succession of two 1D interpolations, with a corner turn in between. In the normal flow of the processing, two more corner turns are then required, one before the range Fourier transform, and one more before the final azimuth Fourier transform. It is desirable for the azimuth transform to come last, as it may need to be iterated several times during autofocus.

The PFA is perhaps the most popular spotlight mode algorithm currently in use. One of its key advantages is the ability to use dechirped inputs, which allows use of “stretch” processing on reception, and reduces the required A-to-D converter speed when imaging sufficiently small scenes. Scene size also is one of its greatest weaknesses for wide-area search missions, since residual quadratic phase errors after the polar reformatting cause significant limitations on the size of the scene that can be imaged at one time using the PFA. This limit on scene size (radius) can be estimated as

$$r \leq 2\Delta r \sqrt{R_s/\lambda} \quad (2.11)$$

Here, r is the scene radius, Δr is the resolution, R_s is the slant range, and λ is the wavelength. For the three cases chosen for the present study, this limitation is summarized in Table 2.5.

CASE #	1	2	3
Scene Radius (km)	2.4	1.5	0.8

Table 2.5. Limitations on PFA scene size for cases in present study.

The RMA was initially introduced for use in stripmap SAR imaging. Like the PFA it also uses dechirped received signals, and has the advantage of being able to use stretch processing for a lower A/D rate. Unlike the PFA it uses a reference

function which performs dechirp only in range, and creates received signals that are stabilized relative to a line at the center of the range swath being imaged. The received signals are thus in the range frequency domain and the azimuth time domain. The RMA uses an azimuth Fourier transform to complete the conversion to the range and azimuth frequency domain, and then a range interpolation to compensate for curvature effects, and then a 2D Fourier transform to complete the imaging. In the normal processing flow, a corner turn is required before the initial azimuth transform, another before the range interpolation, and another after the final range Fourier transform and before the final azimuth Fourier transform.

Because the RMA compensates fully for curvature effects, it is more suitable than the PFA for wide-area SAR imaging. In this role, however, it has another limitation not always experienced by the more restricted PFA but due to the same dechirp-on-receive processing that they both share. This is due to what is called “range skew,” or “residual video phase” after the dechirp receiver processing. When it becomes significant, compensation for this effect for large scene sizes requires an added Fourier transform, weighting, and inverse transform in the receiver prior to the image formation algorithm. The limitation on scene size where this becomes important can be estimated as

$$r \approx \Delta r c / \lambda \sqrt{\gamma} \quad (2.12)$$

Here, all parameters are as defined just above, except that γ is now the transmit chirp rate. For the cases of present interest, the relevant parameters and resulting scene size are summarized in Table 2.6. As it happens, in this case the relevant scene sizes for this phenomenon to be important are just slightly larger than those already given in Table 2.5.

Case #	1	2	3
Wavelength (cm)	3	1.5	0.75
Chirp (MHz/ μ s)	4.9	16.4	49.3
Scene Radius (km)	2.8	1.5	0.9

Table 2.6. Scene sizes for appreciable range skew for the PFA and RMA.

The CSA is similar to the RMA in its use of reference signals stabilized relative to a line at the center of the range swath being imaged, but does not use range dechirp for this purpose, and so requires a higher A/D rate for fine resolution range imaging, but does not carry the burden of residual video phase correction in the receiver unless based on a re-chirping of signals received via dechirp receiver processing.

In this regard the CSA is very similar to the fundamental stripmap SAR method chosen for primary analysis in the present investigation. Like the basic stripmap approach, the CSA requires both forward and reverse Fourier transforms in both the range and azimuth dimensions. Unlike both the RMA and the basic stripmap approach, the CSA involves no range interpolations and only approximately compensates for curvature effects.

The latter feature limits the sizes of scenes which can be imaged. However, for the cases of present interest this limitation is not expected to be a limiting factor. Although the CSA is very similar to the basic stripmap imaging mode in terms of its basic computing operations, it requires a larger number of corner turns (three vs. one) to execute these operations in the normal flow of the processing.

Within their respective limits of applicability, and neglecting inefficiencies due to factors such as excessive pulse length relative to the size of the scene imaged, or

zero padding required for use of fast Fourier transform methods when the number of range or azimuth samples differs significantly from a power of 2, the basic computing requirements of all of the above spotlight imaging methods, and also for the basic stripmap method, tend to be quite similar. This can be understood in part from noting that the total number of Fourier transforms plus interpolations required for all of these methods are the same (four), and in part from the following “rule-of-thumb” formulas for estimating the number of operations required to perform these functions.

$$OP's_{cfft} \cup 5N \log_2(N) \quad (2.13)$$

$$OP's_{int} \cup 5N_{out} L_{in} \quad (2.14)$$

Here equation 2.13 provides an estimate of the total number of (real) operations for a complex Fourier transform of N complex samples, and equation (2.14) provides an estimate of the number of real operations for interpolating N_{out} complex samples with an input interpolation basis of L_{in} samples. For a nominal case of relatively small scenes with $N \approx N_{out} \approx 1024$ and also $L_{in} \approx 10$ the costs of each basic operation are estimated to be about the same. Accordingly, for this specific case, the costs of all of the different SAR imaging methods addressed above (including stripmap) are also estimated to be about the same.

This point has been partially verified by an independent study (published by ERIM) of the computing requirements for the PFA, RMA, and CSA approaches for the very same conditions outlined above. Stripmap imaging was not included in the study, apparently because the scene sizes of interest were small. Including additional details of each different algorithm, the total number of operations per output pixel for the three different spotlight algorithms were as estimated in Table 2.7.

Method	PFA	RMA	CSA
Operations/pixel	280	230	235

Table 2.7. Operations per output pixel for three different spotlight SAR imaging modes, for specific conditions given in the text.

As indicated in the table above, in practice, there will always be additional minor processing steps, also including final interpolation from the slant range plane to the ground range plane). In reality there will also be various inefficiencies in the implementation of the imaging processing, which may accumulate to as much as a factor of a few overall. Oversampling in range and azimuth will also increase the number of required operations per output pixel.

There can also be very major differences in the complexity of the front-end receiver functions. Except for the issues of A-to-D conversion and residual video phase compensation, already discussed above, we will not consider these front-end receiver functions in this study, the principal reason being that although they may require appreciable computing, and bandwidth, they do not generally require large data rearrangements of the sort needed for the corner turns in the image formation processing.

From that last perspective, it also can be seen that the stripmap image formation mode chosen for the present study is actually the least demanding in terms of the total amount of large-scale data rearrangement required for its execution. It requires only one large corner turn, whereas the various spotlight imaging methods reviewed above nominally require three. Accordingly, any potential benefits from high speed crossbar interconnectivity identified in the next subsection may be even greater for other SAR imaging methods, such as those outlined above.

2.4. Processing hardware assumptions

In addition to the ADC issues already discussed, we have briefly examined other hardware technologies relevant to the design of a preliminary architecture for the future use of E/O interconnects for SAR image formation processing for the above three cases. Two issues were of concern: (1) backplane bus bandwidths for the links to/from and through an E/O crossbar switch, and (2) processor or multi-processor card operating speeds. As already discussed, in both cases we were mainly concerned with technologies which could be expected to become available on time scales consistent with the E/O crossbar technology itself.

The current 32-bit wide PCI Bus offers a bandwidth of 132 MB/sec at 33 MHz. The 66 MHz 64-bit wide version is expected to be available in the very near future, providing up to 533 MB/sec. The VME64 Bus offers a bandwidth of 80 MB/sec, while two-edge 2eVME64 has reached data rates of 160 MB/sec. Existing VME320 offers a bandwidth of 320 MB/sec and is projected to be capable of operating at as high as 533 MB/sec without errors. Estimated speeds of future generations of VME Bus are more than 1 GB/sec by the year 2000. On the above basis, we have made a conservative assumption for the bus speed of 400 MB/sec which is expected to be well within the range of both fast-wide PCI and also the next generation of VME beyond VME320.

We have also reviewed available fixed-point and floating-point digital signal processors (DSPs). The current generation of Texas Instrument's C67xx series offers 1 GFLOPS and TI envisions its next generation to be 3 GFLOPS by year 2000. Quad C6701 cards currently offer 4 GFLOPS using a single-slot 6U VME board. The SHARC processor from Analog Device offers 120 MFLOPS, and a single 6U VME board with 24 SHARCs providing up to 2.88 GFLOPS is also

available. Based on the above, we have assumed a sustained throughput of each processing unit to be up to 1.6 GFLOPS.

2.5. Summary of analysis results

With the prior material as background and orientation, the analysis of potential E/O interconnect technologies relative to the chosen SAR imaging applications can now be fairly easily summarized and explained.

Table 2.8 summarizes some of the key factors for the range compression which must be performed on each received radar pulse, and consistent with the PRF.

The number of complex (I and Q) samples per pulse is determined by the transmitted bandwidth, the received signal duration, and the oversampling. For use of fast Fourier transform (FFT) methods this number must be zero-padded up to the next higher power of 2. It also is possible in principle to perform the FFTs on multiple subsegments of the entire received sample train, and thereby perhaps to reduce the total processing burden. Optimization of such an approach depends on the actual transmitted pulse duration, for which the present values are only initial estimates. Therefore, we have not considered that approach here. This is one of several sources of potential “inefficiency” in the resulting computing requirements.

Case #	1	2	3
B(MHz)	246	492	987
T_p (μ s)	50	30	20
ΔR_s (km)	51.4	25.1	12.4
T_{rx} (μ s)	393	197	103
$B \times T_{rx} \times O/S$	1.16E5	1.16E5	1.22E5

FFT Length	1.31E5	1.31E5	1.31E%
OP's/CFFT	1.11E7	1.11E7	1.11E7
CFFT's/Pulse	2	2	2
N _{out}	8.43E4	8.23E4	8.16E4
L _{in}	3	3	3
OP's/INT	1.26E6	1.23E6	1.22E6
Total OP's	2.36E7	2.36E7	2.36E7
PRF	334	669	1338
Total GOPS	7.9	15.8	31.5
Processors (PE's)	6	12	23
Storage/PE (MB)	6	6	6
Output/PE (MB/s)	375	367	380
Polarities	1	2	3

Table 2.8. Initial summary of basic range compression processing parameters.

In interests of parallelization at the highest level, it is assumed that the processing of different polarization channels will be done completely separately. The number of assumed polarization channels in each case is listed in the table, but has no effect on any other values listed. As previously discussed, it may be desirable to eventually bring the different polarization channel outputs back together for noncoherent averaging for speckle reduction purposes.

The processing consists mainly of a complex forward FFT, complex multiplication by a range compression reference function, an inverse FFT, and then interpolation (INT) to remove the range oversampling. Only the FFT's and interpolations are counted here. Additional minor computing costs are expected to be well within the margin of "inefficiency" already discussed. The number of operations required to range-compress each radar received signal, times the PRF, gives the

total computing speed required. Since this is greater than the computing speed of any single Processing Element (PE), the job (including inputting of the data, and outputting the results) must be time multiplexed over a larger number of PE's.

The number of processors (PE's) selected above for each case is designed to allow only a very minimal (20%) overhead for both input and output communications. Inputs are needed to load the data, and outputs must then be transmitted to the PE's which will perform the azimuth compression. This has been done assuming no blockjng or contention at all. As more fully described below, it will be the job of a multi-channel E/O crossbar switch to ensure that this vision can be realized. The resulting output bandwidths are thus 10-times the basic bandwidths needed to transmit the output data from each PE, since it will be done in only 10% of the total time available. These output bandwidths are based on 32-bit complex data. The input bandwidths require a higher sampling rate (due to the oversampling), but are assumed to involve only 16-bit or lower resolution complex data. As also indicated in the table, since they are almost continually streaming data in and out, one received radar pulse at a time, the total storage required for each range compression PE is fairly small.

A related summary of the azimuth compression processing is given in Table 2.9. Here, the number of cross-track samples per synthetic aperture actually varies somewhat with the slant range, but it is shown in the table that for the cases of present interest this variation falls (except at shortest ranges in Case # 2) within the range of FFT sizes already required at maximum range for zero-padding to the next higher power of 2. Therefore, again with some "inefficiency," a single, zero-padded, synthetic aperture FFT size can be assumed for all different range bins.

Case #	1	2	3
Min. Pix./Az. x O/S	4.53E3	7.32E3	1.02E4

Max. Pix. /Az. x O/S	6.85E3	9.51E3	1.24E4
FFT Length	8.19E3	1.64E4	1.6E4
OP's/CFFT	5.32E5	1.15E6	1.15E6
PGA Iterations	2	3	4
# CFFT'S	4	5	6
Range Curvature	14	21	29
OP's/INT	5.7E5	1.72E6	2.38E6
Total OP's/line	2.19E6	7.45E6	9.26E6
Lines/Pulse	8.43E4	8.23E4	8.16E\$
Total OP's/Aperture.	1.85E11	6.13E11	7.56E11
Time (s)	16.6	12.5	8.35
Total GOPS	11.1	49.0	90.5
Overheads	2.4	2.4	2.4
Final GOPS	26.6	118	218
PE's	17	74	136
Lines/PE	4.96E3	1.11E3	600
Storage/PE (MB)	800	600	400
Output/PE (MB/s)	7	3	3

Table 2.9. Summary of basic processing factors for azimuth compression.

Since multiple azimuth compression PE's are clearly required, it is assumed that each PE will deal with a different range swath. The number of range lines that can be assigned to a given azimuth PE is indicated in the table. The input data from the range compression PE's will have been distributed to these azimuth PE's using the E/O crossbar switch, as assumed above, and more fully discussed below.

Prior to autofocus, the main part of the processing consists of a complex forward FFT, complex interpolation (INT) over the range curvature extent with an azimuth

transfer function weighted intermediate, and an inverse FFT to recover the final complex image. Another key part of the processing is autofocus, which requires a few or several iterations of the final FFT, as well as other functions to estimate the residual phase error. The required number of autofocus iterations depends in part on the accuracy of the initial time/position data provided to the receiver front-end by feedback from GPS and the on-board INS unit(s). For existing systems it has been claimed that the PGA can converge in as few as 2 iterations for final resolutions on the order of one meter or better. At the higher resolutions anticipated in here this may no longer be true unless there are proportionate improvements in the accuracy of the feedback to the receiver front-end. We have assumed some progress in this regard over the time scales envisioned, but have also assumed a moderate increase in the required number of autofocus iterations, as indicated in the above table.

The total number of operations for each aperture is the number for each imaged range line times the number of range lines per radar pulse. The total rate at which this must be done is then inverse to the time required to accumulate the multiple received pulses needed to form the aperture. Due to the assumed 50% overlap of consecutive apertures, there is an added factor of 2 overhead, and we have included another factor of 1.2 overhead for multiple additional minor operations, such as converting from complex to real, removing the 1.2 oversampling in azimuth, converting from slant range to ground range, and other minor factors in the azimuth image processing itself. As already mentioned, some of this overhead can also be compensated by more tightly optimizing the FFT lengths for the different range intervals. A 10% overhead for input and output communications is also assumed to have been included here, and will be discussed below.

Also indicated in the table is the estimated storage requirement for each azimuth compression PE. This required storage is appreciable, since we are assuming that

all required azimuth transfer functions needed for the range curvature interpolation will have been precomputed and stored. The total amount of storage per azimuth PE is proportional to the number of range lines imaged per PE, number of range curvature bins, and number of samples per aperture. In practice, this can be reduced somewhat by re-using transfer functions over adjacent range bins, since the range-dependency of the azimuth transfer functions is slowly varying. Some reduction of the required storage on this account has been assumed.

The assumed 10% overhead for both input and output communications is based in part on noting that there are generally from 3 to 6 more azimuth processors than range compression processors, so the time spent by each azimuth compression PE in receiving data can be that much less than the time spent by each range compression PE in transmitting its data to multiple azimuth compression PE's. In addition, the final output rates of each azimuth compression PE are less than one percent of a single 400 MB/s output channel. Therefore, only one or a few output channels, ordered by range, could be shared among multiple azimuth PE's with negligible output communications overhead for each azimuth processor involved. However, this again depends on the availability of a non-blocking crossbar switch to permit the multiple assumed and time multiplexed data transfers to be realized in practice.

Figure 2.10 provides a schematic of the envisioned computing architecture on which all the above is based. The heart of the architecture is a large crossbar switch, presumably based on the E/O technology currently under development in the DARPA VLSI photonics program. As described above, the availability of such a switch would allow SAR image formation, even for the very demanding cases addressed in the present study, to be performed in real-time, and generally with a 20% or lower overhead in required total computing power due to input, inter-processor, and output communications delays.

As indicated, it is envisioned that incoming data will be distributed to the multiple range PE's using a single input channel (per radar polarization), which is time-switched over the multiple range PE's at a rate equal to the radar PRF. The output data from each range compression PE will then be parted out to different azimuth compression PE's, with a switching rate equal or greater than the PRF times the number of azimuth PE's. The azimuth PE's then time share a single (or more) output channels, with a switching rate appreciably lower than the above. The basic system would be replicated for multiple polarization channels.

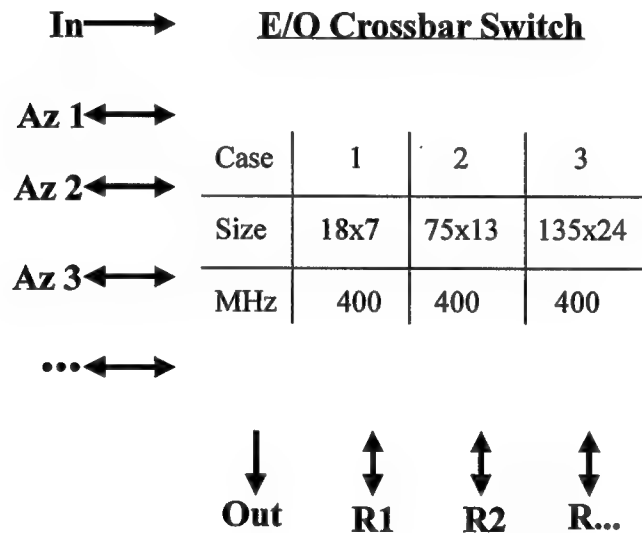


Figure 2.11. E/O crossbar switch for SAR image formation processing.

3. AUTOMATIC TARGET RECOGNITION (ATR)

Automatic target recognition is a demanding area of high performance computing for defense applications. It is therefore an excellent candidate application to address the potential benefits from using high speed E/O interconnect technologies. ATR requires very high processing and inter-processor communications performance, involving a succession of local analyses of globally-distributed data. Ability to distribute, and then redistribute, global data among the local processors is paramount to optimal ATR performance.

Here, an initial investigation is done in terms of evaluations of the tradeoffs between required ATR computing power versus available inter-processor communications bandwidth and switching connectivity. The results from these analyses are designed to provide insight into promising application areas and possible directions for future development of E/O interconnect technology for future defense computing applications.

Three fundamental approaches to ATR are well-known. The first is simple template matching. It requires a huge amount of memory to store templates of all target types and configurations of interest. Generating these data is also problematic. Another approach, used by MSTAR program, is to store radar 3D target data, from which templates are then generated on-the-fly. This avoids the template storage problem, at the cost of generating the templates on the fly, and any performance limitations caused by computer-generated templates. Also, it still is fundamentally a template matching approach, and consequently less efficient than more advanced methods for statistical pattern matching, as used in speech and other areas. The third approach is to use more sophisticated statistical modeling and feature extraction methods to model the underlying distributions of target and clutter properties, typically based on Artificial Neural Network (ANN) or Gaussian Mixture Model (GMM) techniques for modeling, and nonlinear analysis and eigenvector techniques for feature extraction. The latter approach can also be based on previously

acquired or synthetically generated target data, but does not require that such data be either stored as such or generated for use during execution of the ATR algorithm.

The latter approach to ATR will be used as the basis of the present study. In particular, we will address a multi-stage statistical modeling approach based on GMM modeling and nonlinear-plus-eigenvector feature extraction techniques, which has shown excellent performance both on prior radar data from the MSTAR program, and also on prior synthetic FLIR target data in real clutter backgrounds. We will examine this statistical model based ATR approach in terms of its stage-by-stage computing and interconnect requirements for varying rates (and spatial resolutions) of input data, and will identify the role that E/O optical interconnect crossbar switching can play in its efficient implementation.

The next subsection gives a preliminary assessment of computing and communications requirements for current and future ATR applications in general. Following that, we will briefly summarize methods (and prior results) of the current statistical approach to ATR. Analyses of computing, data storage, and timing and inter-processor communications requirements for distributed multi-processor real-time implementations of the approach are then provided. In Section 4 we discuss the potential impacts from E/O interconnect technologies in real time implementations of ATR processing using this approach, and also with reference to the synthetic template matching approach of the MSTAR program.

3.1 Initial Study

In this project, we initially performed a very preliminary analysis of current trends in ATR processing, and of the potential implications for E/O interconnect technologies in general. This preliminary study did not include a stage-by-stage analysis of specific processing details and potential hardware implementations, but rather estimated overall

trends and tradeoffs between the required computing power versus communication bandwidth for prior, current, and envisioned future ATR processing methods.

The results from this preliminary analysis were provided to the then-current DARPA COTR (Dr. A. Hussain) at a very early stage in the present effort. Even as limited as they were, Dr. Hussain has stated that these preliminary findings were of critical and irreplaceable importance for his defense of both the Free Space Optical Interconnect Applications (FSOIA) program, and also the follow-on VLSI Photonics program.

ATR, in general, is a multi-stage screening process. Earlier stages use relatively simple processing of lower resolution (subsampling) data to detect potential target locations. This early screening requires a relatively small amount of computing per pixel, but is required to address a very large number of pixels overall. Subsequent processing stages focus on areas detected to be of interest from prior stages, and use a higher amount of processing per pixel as the number of pixels of interest (and also the number of candidate target hypotheses) are reduced.

This general structure is shown in figure 3.1, using three stages. The first stage finds regions of interest using relatively simple and less expensive methods on coarse resolution data. These anomalous regions are then passed on to preliminary detector stage. This stage also works with coarse data. It detects target-like locations using some decision metric such as likelihood ratio and also provides a ranked hypothesis list (soft decision index) to the last stage. Since a very small fraction of total data set needs to be processed further, this last stage uses more sophisticated methods and fine resolution data. At this final stage, target detection/classification is done using target models. These target models can be simply target templates, 3D radar models, or statistical models of target distribution functions, such as GMMs or ANNs.

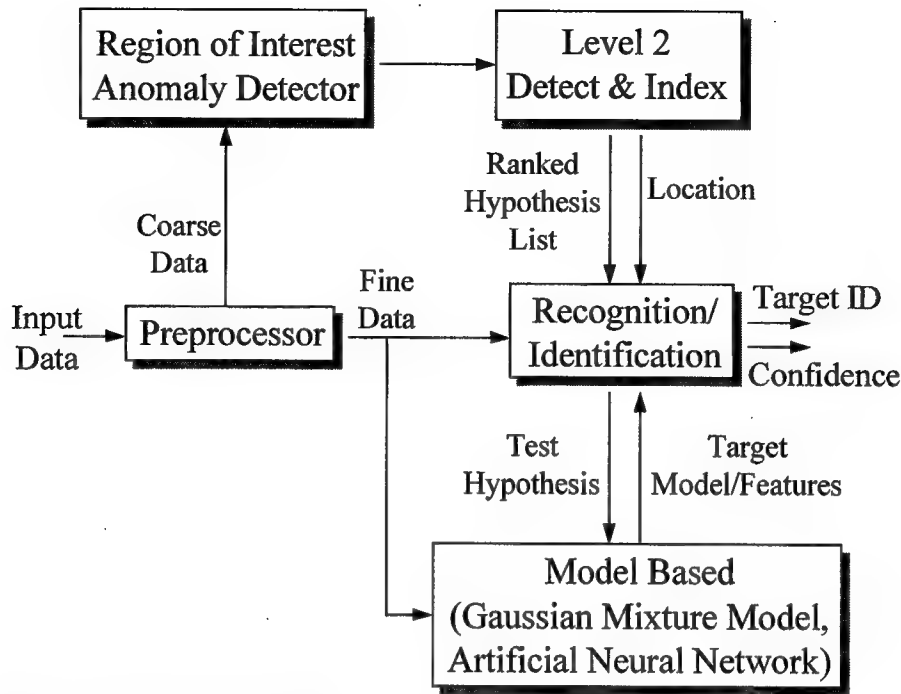


Figure 3.1 A basic structure for ATR processing.

A basic summary of the anticipated computing versus communications requirements for prior, current state-of-the art, and future advanced technology ATR applications is given in figure 3.2. The label “STARLOS” is used to designate prior methods developed and used by Sandia National Laboratory (SNL), involving a multi-stage algorithm using template matching and indexing. For the analysis of this approach we have assumed a pixel rate of 2×10^6 per second and only 6 different types of targets present in the data set. Total operations per second (OPS) for STARLOS are estimated at 3×10^{11} . The label “Tier II” is used to designate the MSTAR approach applied to data from the Tier II radar. This is a template matching/indexing and model based approach. It is analyzed with a ten times higher pixel rate than STARLOS, and the total number of target types are assumed to be 30, for an estimated total computing burden on the order of 10^{12} OPS. The future generation was considered based on a future trend from template-based to statistical model-based ATR and increasing sensor resolutions and search rates. Input pixel rate was assumed to be 2×10^8 per second, and the required computing power was estimated to be on the order of 3×10^{12} OPS.

As indicated in the figure, the rate of increase in required raw computing power is projected to decrease somewhat as processing methods become more sophisticated, but accompanied by greater demands for higher resolution floating point computing, and also to be outstripped by the rate of increase in required input/output and inter-processor communications.

	Prior (STARLOS)	Current (Tier II)	Next Gen (TBD)
Pixel Rate	2×10^6 /s	2×10^7 /s	2×10^8 /s
Targets	6	30	30
Methods	Multi-stage template match w/index	Template match/index w/ model based	Multi-stage model based w/ index
Int MACS	3×10^{11}	10^{12}	$\leq 10^{12}$
FP MACS	3×10^8	10^{11}	2×10^{12}
Total OPS	3×10^{11}	10^{12}	3×10^{12}

Figure 3.2 Comparison of computing power required by prior, current and future ATR methods.

A rudimentary tradeoff of bandwidth versus actual required computing power, now in terms of MACS (Multiply-accumulate per second), is shown in Figure 3.3. This depicts the fact that if insufficient or even marginally sufficient bandwidth is provided then the required amount of computing power can be much higher than the values in Figure 3.2, since the processors are under-utilized due to communications delays. Figure 3.3 is only a very qualitative depiction of the case: the specific bandwidth requirements are based on the input data rates from Figure 3.2, and do not account for additional inter-processor communications, nor for specific algorithm and implementation details. Yet, for those who see the future needs of ATR technology mainly in terms of increased computing power, it is a powerful message. The processing power actually required for future ATR implementations is critically tied to the inter-processor bandwidth that can be provided, a problem for which E/O interconnect technologies may provide the requisite solution.

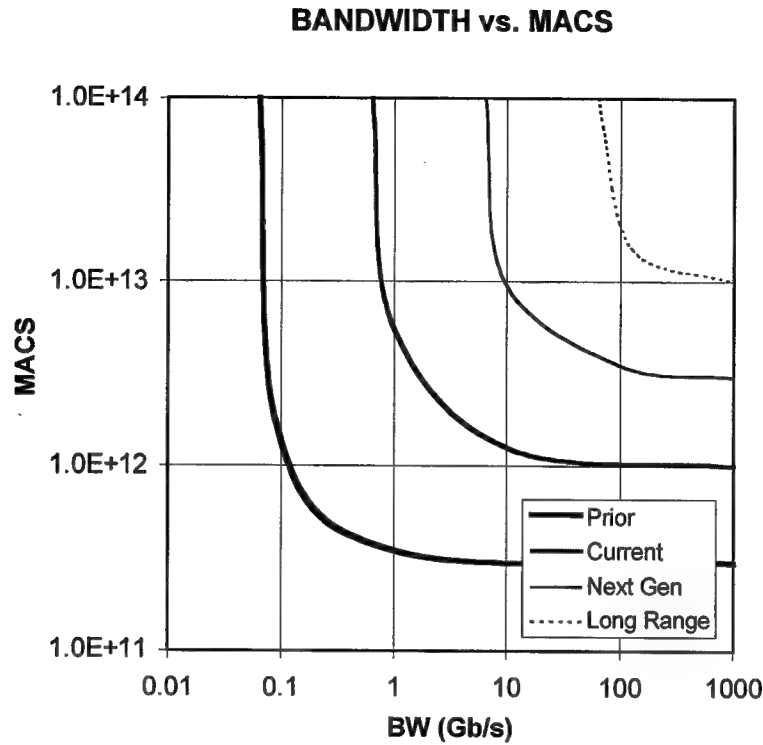


Figure 3.3 Prior, current and future computing versus bandwidth.

While the trades shown in Figure 3.3 are highly idealized preliminary results, more detailed analyses of the interplay between communications bandwidth and required processing power will be studied in the following subsections. These studies will account for actual implementation details including: (1) the number of independent processors, (2) the amount of local memory per processor, (3) the amount of global memory, and (4) the overall architecture of the inter-processor communications assets.

These preliminary findings outlined above were provided to Dr. Hussain as a summary view graph, shown in Figure 3.4, which he was able to use in defense of both the Free Space Optical Interconnect Applications (FSOIA) program, and also the follow-on VLSI Photonics program.

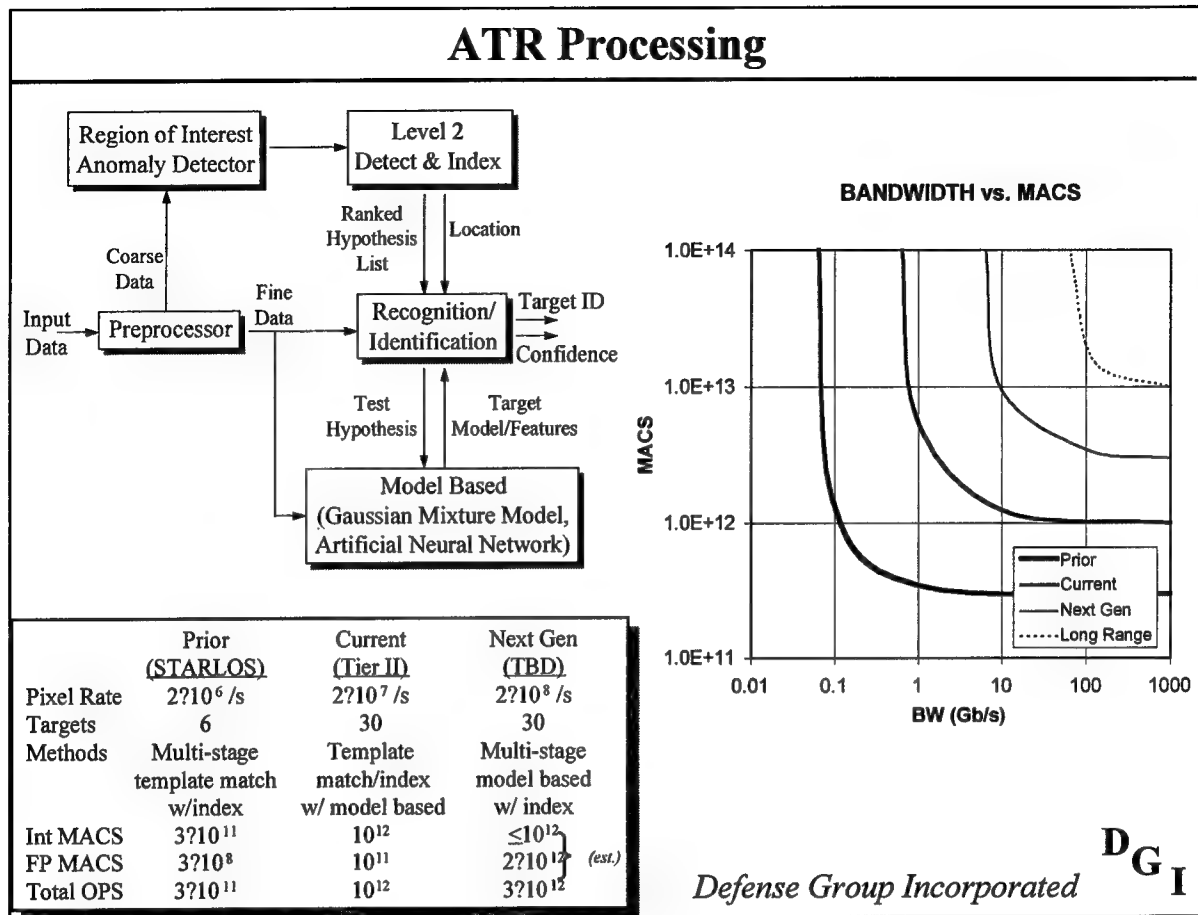


Figure 3.4 Prior, current and projected future computing versus bandwidth trades for automatic target recognition.

3.2 Detailed Study

As discussed above, the conventional template matching approach to ATR is simple but inefficient, and expected future trends are from template-based to statistical-model-based ATR. Therefore, we chose to explore ATR processing using a statistical modeling and feature extraction approach. This approach includes nonlinear preprocessing, efficient statistical modeling of target and clutter distribution functions, eigenvector feature extraction, and staging the processing as a succession of screenings of increasing sophistication as uninteresting prior inputs are screened out.

The potential performance of these methods has already been demonstrated using synthetic FLIR target data in real backgrounds, and an initial set of MSTAR SAR data. Some examples of these data can be found at www.ca.defgrp.com. Prior performance details are included below. Details summarized below also include designs for real-time multiprocessor implementations of the basic approach, and the potential impacts of E/O interconnect technologies for such implementations.

3.2.1 Overview of Basic Approach and Summary of Prior Results:

The basic processing structure is shown in Figure 3.5. This depiction shows a total of four sequential on-line processing stages: (1) a clutter anomaly detection, (2) an initial target versus clutter detection and optional soft-decision indexing, (3) a secondary target versus clutter detection and soft-decision indexing, and (4) a final target versus target/clutter discrimination and classification stage. Also shown are the on-line feature extraction stages and their off-line nonlinear and linear feature selection and distribution function model building stages.

The first on-line stage is a scene-adaptive anomaly detector, that analyzes the statistics of each input scene, and determines those pixels (and surrounds) that are most anomalous. This initial stage does not depend on target details, new targets, or altered signatures; it finds anomalies relative to an adaptively determined clutter distribution function model.

Initial results from this stage, based on an initial MSTAR SAR Public Set of target-in-background data, are shown in Figure 3.6. The figure shows false alarm rate (FAR) versus percentage of falsely rejected targets (PFR) at this stage. As indicated by an “arrow”, FAR is reduced to less than 5×10^{-4} at PFR of 0.3%, which means that the input data to the next stage can be reduced by a factor of more than 400.

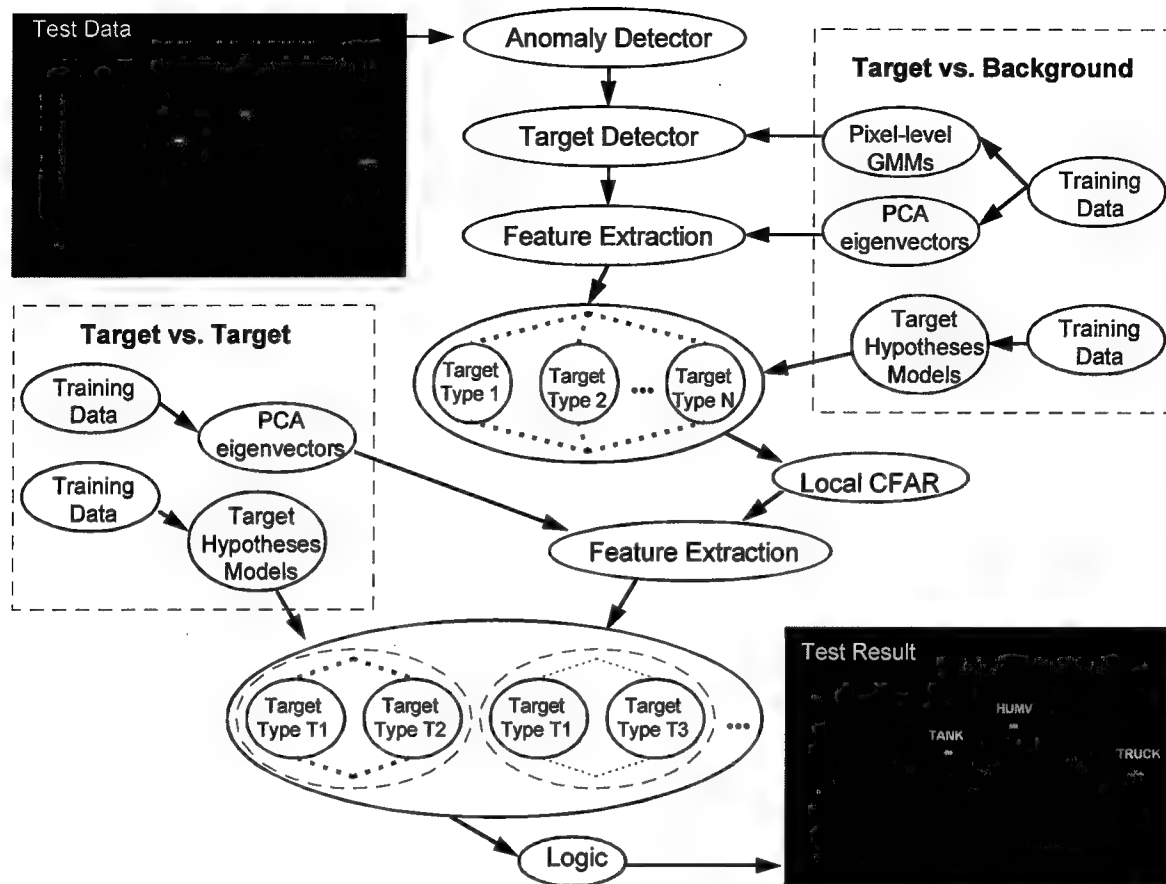


Figure 3.5 Most basic multi-stage ATR processing chain, showing both on-line and off-line components.

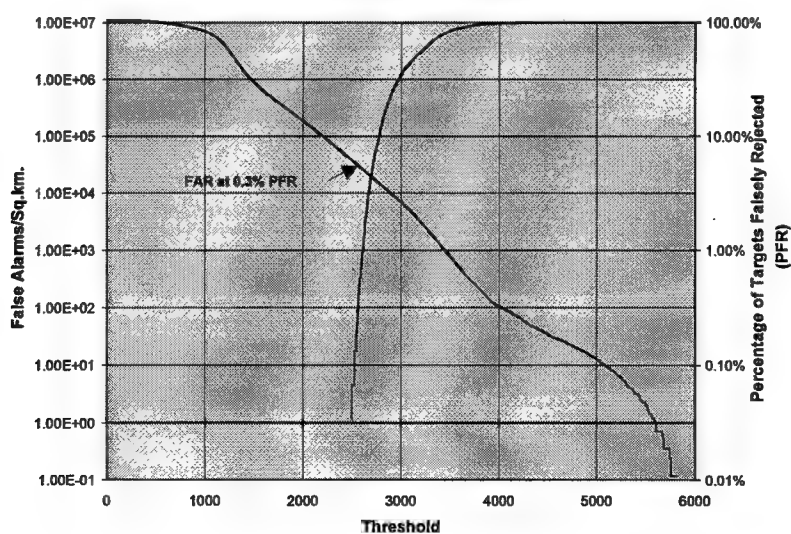


Figure 3.6 First stage FAR/PFR versus detection metrics based on MSTAR data.

The second on-line stage is a preliminary target detector, which uses target statistics, as well as the clutter statistics determined above, in analyzing the selected pixels and surrounding regions, but without spatial correlations of target or background statistics. Each object of interest is analyzed using specially designed Gaussian mixture models to represent the target hypotheses, and an adaptive clutter model derived above for the non-target hypothesis. Indexing (tentative initial ranking of hypotheses) is supported at this stage, and can optionally be used to focus and reduce the cost of subsequent processing.

Figure 3.7 shows the probability of false alarm (PFA) per input pixel versus the probability of false rejection (PFR) of valid targets at the output of this stage, both for a FLIR data set, and also for the MSTAR SAR Public Data Set. For the SAR data, an additional 25-fold reduction of inputs needing further analysis is achieved at 0.3% PFR. These reductions of the inputs that need to be passed on to the next processing stage have a key role in determining the required inter-processor communications loads.

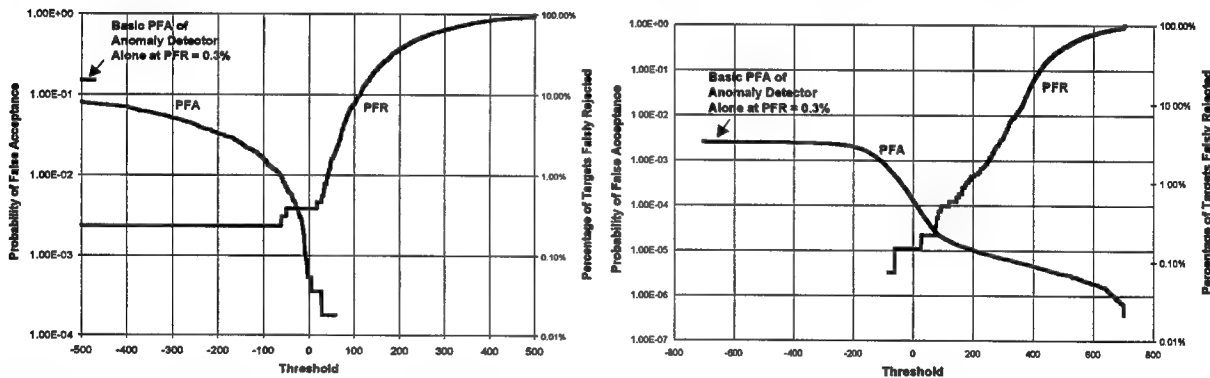


Figure 3.7 PFA/PFR for initial target detection. FLIR at left, MSTAR at right.

Figure 3.8 shows the corresponding initial target classification results at this processing stage, both for a three-target FLIR data set, and also for a three-target SAR data set.

Target Type	Classification		
	Tank	Truck	Humv
Tank	81.20%	8.30%	10.50%
Truck	10.40%	77.70%	11.90%
Humv	9.20%	11.40%	79.40%

Target Type	Classification		
	BMP-2	BTR-70	T-72
BMP-2	88.80%	1.40%	9.80%
BTR-70	8.20%	91.30%	0.50%
T-72	9.10%	0.90%	90.00%

Figure 3.8 Classification/indexing results at initial target detection stage. FLIR results at left, SAR results at right.

The third on-line stage is a specially designed extension of principal component analysis (PCA) to derive specific features that are most probative for distinguishing background clutter from the various target hypotheses of interest, followed by parallel analyses of each hypothesis (or any indexed subset) based on full covariance GMM techniques. This stage supports “Next-stage” hypothesis indexing (instead of hard decisions). It further reduces the objects of interest by a factor of approximately 5.

The final on-line stage processes the subset of data selected above, and compares these data with the most likely prior selected hypotheses. This is done by a succession of pairwise comparisons, each comparison using nonlinear preprocessing followed by the linear extraction of a subset of features best designed to distinguish between the members of each hypothesis pair, and GMM-based distribution functions for each pair member. This stage yields final decisions. Figure 3.9 shows the final PFR versus FAR results for both data sets described above. For both data sets, less than 1 false alarm per sq. km. is achieved at the PFR of about 1%.

Also shown in Figure 3.10 are the final stage target classification results, both for the FLIR and SAR data sets. These results based on both data sets are obviously very good.

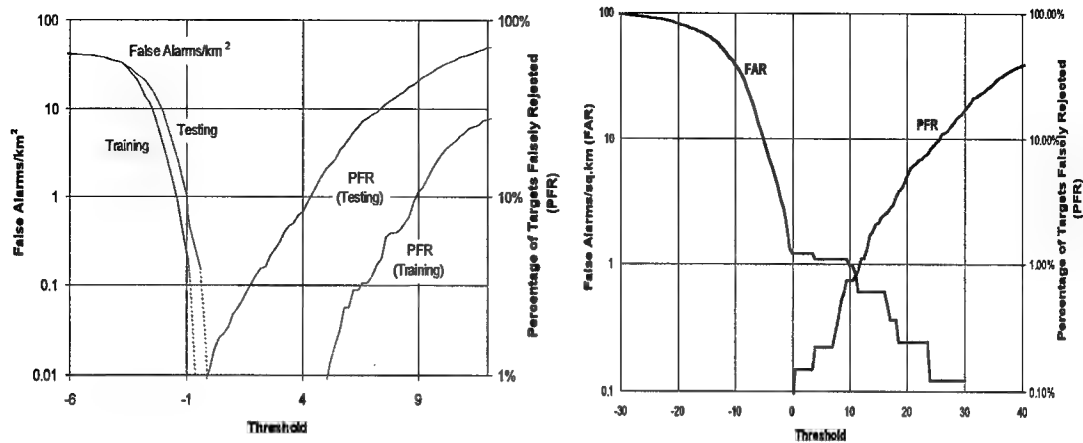


Figure 3.9 Final stage PFA/PFR results for FLIR data on left and SAR on right.

Target Type	Classification		
	Tank	Truck	Humv
Tank	98.15%	0.46%	1.39%
Truck	0.47%	97.67%	1.86%
Humv	0.46%	0.23%	99.30%

Target Type	Classification		
	BMP-2	BTR-70	T-72
BMP-2	98.12%	1.54%	0.34%
BTR-70	1.53%	98.47%	0.00%
T-72	1.38%	0.52%	98.10%

Figure 3.10 Final classification results for FLIR data on left and SAR data on right.

Several off-line support processes are also depicted in Figure 3.5. Since these off-line processes are not an issue for real time implementation, we provide only a brief overview.

Early-stage off line processes accumulate uncorrelated target statistics for preliminary target detection and generate nonlinear input transforms and higher-order uncorrelated GMMs by Expectation Maximization (EM) training. The second off-line process accumulates correlated target and background clutter statistics and thus derives the feature extraction parameters for use in the classification stage(s). The third process creates GMMs with full covariance matrices, or single-hidden-layer ANNs, for use in analysis of the resulting features. Related off-line methods are used for the final processing stage.

3.2.2 Technical Description of the Approach:

In addition to the overall summary provided above, a further technical description of the approach is necessary to describe the basis for the following calculations of computing power, memory, and communication bandwidth.

First-Stage Anomaly Detection: This processing stage identifies image regions that are statistically different from the input clutter. The anomaly detector models input clutter probability density functions (PDFs) via an iterative (two-step) process that includes spatially-adaptive detrending to reduce clutter nonstationarity. Anomalous regions are detected based on their dissimilarity to local clutter statistics, as represented by the estimated clutter PDF.

The basic processing includes: (1) local mean estimation and subtraction; (2) local variance estimation and normalization of the residual; (3) histogramming, followed by a first-pass detection of anomalous pixels; (4) re-estimation of local mean omitting the anomalous pixels; (5) re-estimation of local variance also omitting anomalous pixels; and (6) final histogramming also omitting anomalous pixels, to give a final determination of the clutter PDF. Local means/variances are computed in annular regions surrounding each pixel of interest and sized to prevent biasing if a target is present at/near the pixel of interest. Effects of possible target presence are further reduced by performing the process in two passes, with anomalous pixels from the first pass not used in estimating means, variances, and histograms in the second pass. A summary of the processing is given in Figure 3.11.

If the parameter y_i represents the mean and variance detrended value at the i -th pixel, and $f(y|\phi)$ represents the corresponding detrended clutter PDF, the anomaly detector can then be based on the log-likelihood metric

$$LL = \ln[f(\vec{y}|\phi)] = \ln \prod_i f(y_i|\phi), \quad (1)$$

where the product in equation (1) is over a target-size rectangular region about each pixel of interest, and is evaluated using an efficient sliding window technique.

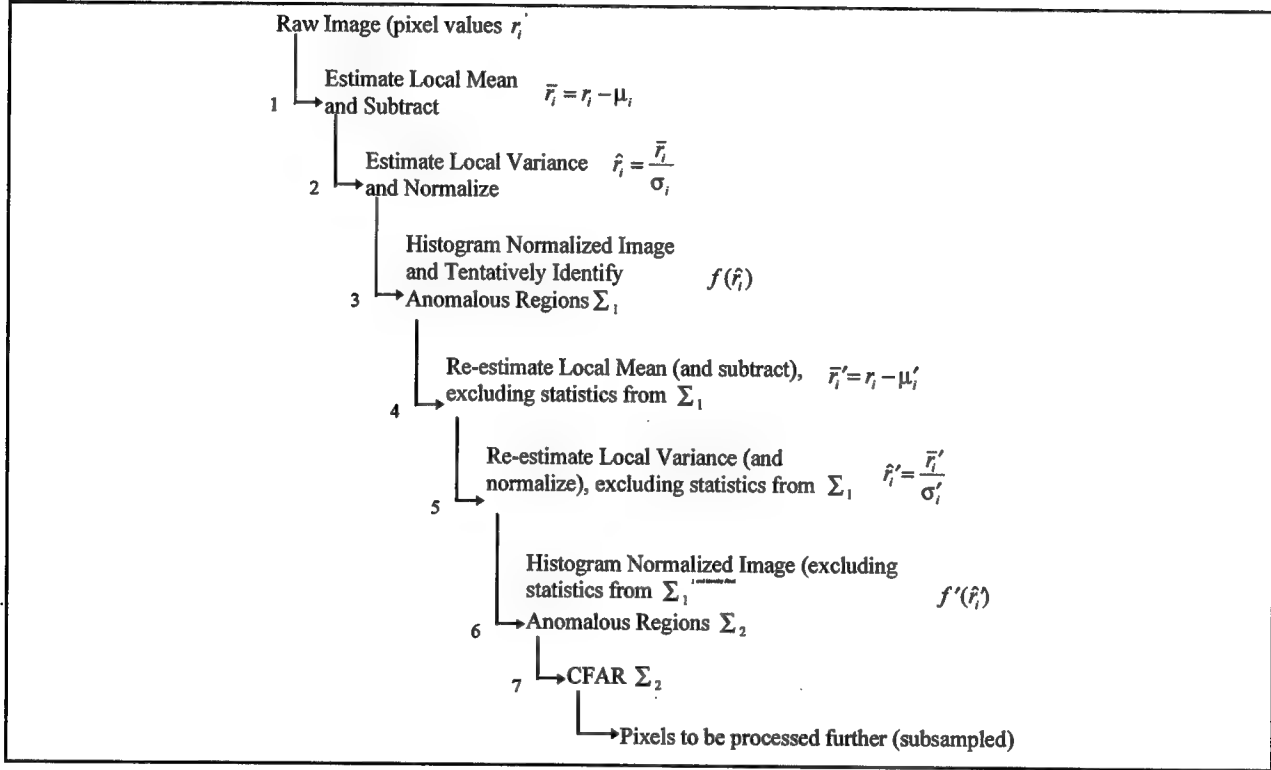


Figure 3.11 Initial processing for scene-adaptive clutter identification and modeling.

Second-Stage Target Detection: This stage uses both target and background statistics in analyzing the pixels and surrounding regions selected from the first stage and is also capable of preliminary indexing. It computes log-likelihood-ratio (LLR) metrics from data in target-sized windows that surround each pixel of interest. LLR is the log ratio of PDFs for two hypotheses:

$$H_0 \text{ -- only background clutter} \quad \text{PDF} = f(\vec{x}|\phi)$$

$$H_1 \text{ -- target near pixel of interest} \quad \text{PDF} = f(\vec{x}|t)$$

The PDFs are modeled without spatial correlations, otherwise the computing at this stage would be too great. The PDF for the null hypothesis is computed as

$$f(\vec{x}|\phi) = \prod_i \{f'(y_i|\phi)(\sigma_i')^{-1}\}. \quad (2)$$

For H_t the PDF of pixel values when a specific target is centered near the pixel of interest is represented as a Gaussian mixture model with N_c centers and is given by

$$f(\vec{x}|t) = \sum_{i=1}^{N_c} p_i f_i(\vec{x}|t). \quad (3)$$

The mixture model is formulated such that each center corresponds to a particular target viewed over a small range of azimuths. A nominal approach has 36 azimuth bins for each target. In developing each $f_i(\vec{x}|t)$ one first determines a template \mathbf{R}_i of those interior pixels most likely occupied by a target of the type and orientation $i \in [1, N_c]$.

One then models $f_i(\vec{x}|t)$ as

$$f_i(\vec{x}|t) \propto \prod_{j \in \mathbf{R}_i} f_{ij}(x_j|t) \propto \prod_{j \in \mathbf{R}_i} f_j(x_j|\phi), \quad (4)$$

where each term in the second bracket is the clutter distribution, as also used in equation (2); and for terms in the first bracket the distributions for the individual pixels are further represented by simple one-dimensional (1-D) GMMs with N_g centers.

Only 5 Gaussian centers are used for the 1-D GMMs. These serve to capture the variations in each pixel's intensity for specific targets with orientations in a given angle bin. The PDF for each target of the type and orientation $i \in [1, N_c]$ and pixel $j \in \mathbf{R}_i$ is given below,

$$f_{ij}(x_j|t) \propto \sum_{k=1}^{N_g} \frac{\alpha_{ijk}}{\sqrt{2\pi}\sigma_{ijk}} \exp -\frac{1}{2} \frac{(x_j - \mu_{ijk})^2}{\sigma_{ijk}^2} \quad (5)$$

Third-Stage Target Detection/Classification: This target classifier stage considers only those pixels (and surrounds) that pass the previous two screenings. Since this is only a fraction of the total data, a large amount of processing per pixel can be afforded. The classifier stage consists of three levels, whose functions are: (1) to extract features from the data about each pixel of interest; (2) to process these feature data through parallel classifiers for the different specific target types; and (3) to make a decision based on the outputs as to whether a specific target is present, or if only clutter is present. Each of these processing levels is described below.

Feature extraction is done as below, where x_i is the i -th component of the input window, v_{ji} is the i -th component of the j -th principal eigenvector of a specifically designed covariance matrix, y_j is the j -th derived feature, and $\langle x_i \rangle$ is an average over all data .

$$y_j = \sum_i (x_i - \langle x_i \rangle) v_{ji} , \quad (7)$$

Irregularly-shaped data window around each pixel of interest (nominally ≈ 700 pixels) is then linearly filtered into nominally about $\approx 50 - 100$ features. A covariance matrix is computed for a data set that is 50% background clutter data that have been passed by first two stages, and 50% target-in-background data from all target types. The resulting features are optimal for distinguishing the targets from those background clutter that pass the initial screenings.

At the present stage, the classifier then consists of parallel channels, one for each target type. Each of these computes either a class-specific ANN output or an LLR metric based on the features \vec{y} computed above. The LLR computed by the classifier for target type i is the logarithm of the ratio of PDFs, based on the feature vectors, for the hypotheses:

$$H_i \text{ -- target type } i \text{ is present} \quad \text{PDF} = f(\vec{y}|t_i)$$

$$H_\phi \text{ -- background clutter is present} \quad \text{PDF} = f(\vec{y}|\phi)$$

For both hypotheses, the PDFs based on the feature data \vec{y} are full covariance GMMs:

$$f(\vec{y}|H_i) = \sum_j \alpha_{ij} f(\vec{y}|H_i, \alpha_{ij}, \vec{\mu}_{ij}, \mathbf{M}_{ij}), \quad (8)$$

where $f(\vec{y}|H_i, \alpha_{ij}, \vec{\mu}_{ij}, \mathbf{M}_{ij})$ is the j^{th} Gaussian PDF with mean vector $\vec{\mu}_{ij}$, covariance matrix \mathbf{M}_{ij} , and mixture weight α_{ij} for hypothesis H_i (or H_ϕ). In our baseline ≈ 40 components are used to estimate the PDFs for each of the hypotheses under test. The LLR decision metric for the target type i classifier is then given by

$$\text{LLR}_i = \ln \frac{\sum_j \alpha_{ij} \frac{1}{|\mathbf{M}_{ij}|^{1/2}} \exp \left\{ -\frac{1}{2} (\vec{y} - \vec{\mu}_{ij})^T \mathbf{M}_{ij}^{-1} (\vec{y} - \vec{\mu}_{ij}) \right\}}{\sum_j \alpha_{\phi j} \frac{1}{|\mathbf{M}_{\phi j}|^{1/2}} \exp \left\{ -\frac{1}{2} (\vec{y} - \vec{\mu}_{\phi j})^T \mathbf{M}_{\phi j}^{-1} (\vec{y} - \vec{\mu}_{\phi j}) \right\}}. \quad (9)$$

Final Target Hypothesis Deconfliction: This stage is a relatively straightforward extension of the above, aimed at even better differentiation between different target types. Inputs here are only those pixels and surrounds already passed through all processing stages above, and already detected as having a target. Since this will be an even smaller fraction of the total data, an even more sophisticated level of processing can be afforded.

The processing here is based on the same general feature extraction methods described above, but now with features specifically designed to differentiate one target type from another. One thus computes (off line) covariances and principal eigenvectors for different ensembles that are 50:50 mixes of different target pairs, and (also off line) full-covariance GMM models for each target type in terms of these target-vs.-target features. It is to be noticed that the off line computing needed to do this is not basically a problem, but there is an added cost in terms of model and feature storage for use in subsequent on-line processing. The computing cost for this stage will be, for each target-versus-target

deconfliction to be analyzed, about the same as each target-versus-background analysis and (firm or tentative) classification addressed above. This can be controlled by limiting the number of comparisons done here based on soft decision metrics from the preceding stage.

3.2.3 Computing Power and Memory Requirement:

We have analyzed computing burdens, data storage requirements, model parameter storage and inter-processor communications requirements for three different input pixel rates: 2×10^6 (STARLOS), 2×10^7 (Tier II) and 2×10^8 (Future Generation) per second. The lowest pixel rate assumes 3x3 sq.ft. input resolution, the second rate has 1x1 sq.ft. resolution, and 0.5x0.5 sq.ft. resolution is taken for the highest pixel rate. Also, the first two cases assume single polarization SAR data, while two-polarization input data was considered for the highest pixel rate. Basic characteristics of these different pixel rates are summarized in Table 3.1

We will discuss the case with a pixel rate of 2×10^7 pixels/sec (comparable to Tier II SAR) in greatest detail. Total operations per second (OPS), memory and bandwidth are also summarized for all three pixel rates at the end of this subsection.

PIXEL RATE (/SEC)	2×10^6	2×10^7	2×10^8
Input resolution	3 x 3 sq.ft.	1 x 1 sq.ft.	0.5 x 0.5 sq.ft.
Number of polarization	1	1	2

Table 3.1 Prior, current and future pixel rate and input resolution.

First-Stage Anomaly Detection: This is the first stage in the processing chain and is required to process all of the input pixels i.e. 2×10^7 pixels per second. Since there are many objects to be processed, relatively less expensive processes are designed at this stage. This includes the use of very efficient sliding window techniques to compute local means/variances and LLR metrics, which drastically reduces the number of operations required at this stage. In this technique, once the column sums for the objects in the first row are computed, then it requires only 5 operations/pixel to compute means of the remaining pixels in the same row, and 4 additional operations/pixel to update the column sums for the pixels in an adjacent row. For the assumed pixel rate, it approximately requires 200 MOPS (million operations per second) for local mean estimation, 240 MOPS for local variance estimation, 10 MOPS for PDF computation and 50 MOPS for computing likelihood ratios. Expensive operations like exponential and natural logarithm can be avoided by using look-up tables (LUTs). Since anomaly detection is done by a two-stage process, the total processing adds up to 1.12 BOPS.

Assuming 4 Bytes per pixel, this stage is required to store 80MB of input data. Since this initial stage is very general and does not depend on target details, new targets, or altered signatures, any off-line processing and model parameters are not involved. From our prior experience with PC-Windows 95/98/NT and HP-Unix implementations of the processing, a local memory of less than 2 MB is required to store the compiled source code of this stage. Interprocessor communication bandwidth is required to be higher than at least 80 MB/Sec in order to maintain real time processing.

Similar analysis was carried out for other two pixel rates also. Although the highest pixel rate has 2 polarization and 0.5x0.5 sq.ft. resolution, this stage processes input data at 1x1 sq.ft. resolution. The following table 3.2 summarizes objects to be processed, OPS and memory requirement of this stage for all three pixel rates.

Anomaly Detector	2×10^6 pixels/sec.	2×10^7 pixels/sec.	2×10^8 pixels/sec.
Resolution used	3 x 3 sq.ft.	1 x 1 sq.ft.	1 x 1 sq.ft.
Objects processed	2×10^6	2×10^7	5×10^7
Pixels to be stored	2×10^6	2×10^7	2×10^8
Data storage	8 MB	80 MB	800 MB
Model parameters	2 MB	2 MB	2 MB
OPS	112 MOPS	1.12 BOPS	2.8 BOPS

Table 3.2 Computing and memory for anomaly detector stage versus pixel rate.

Second-Stage Target Detection: This stage processes only those pixels which have passed through the anomaly detector stage. As shown in Figure 3.6, the anomaly detector reduces the number of objects to be processed by a factor of approximately 400 at the PFR of about 0.3%. Total objects to be processed at this stage reduces to 5×10^4 per second. Since this is a preliminary target detection, we are able to achieve very good results even with a coarse resolution of 4x4 sq.ft. In a nominal approach this stage has 36 azimuth bins for each target, and we are assuming 30 different target types present in the data. Surrounding pixels per object orientation are approximately 35, and 25 operations are required to evaluate 5 Gaussian centers per surrounding pixel. Thus 35x25 operations are required to evaluate the PDF for the target hypothesis per object per target-orientation per target, which adds up to a total of $30 \times 36 \times 35 \times 25 \times 5 \times 10^4$ operations per second to compute target-present PDF. Since this also is an iterated two-pass process, we need a total of approximately 100 BOPS in this stage.

Each object is passed to this stage along with its surrounding pixels within a target size box of about 100 pixels. Thus this stage is required to store 20 MB of input data. The clutter PDF normalized by its local variance is also passed to this stage from the anomaly detector stage which accounts for another 20 MB. Also 2 MB of local memory per processor is required to store GMM model parameters (12 parameters per surrounding

pixel per target-orientation per target) and again an additional 2 MB to store the code. The analysis of this stage is shown in the following table 3.3 for all three pixel rates.

Target Detector	2×10^6 pix/sec.	2×10^7 pix/sec.	2×10^8 pix/sec.
Resolution used	3 x 3 sq.ft.	4 x 4 sq.ft.	4 x 4 sq.ft.
Surrounding pix/tgt	178	100	100
Objects to be processed	1.5×10^4	5×10^4	6.25×10^4
Pixels to be stored	2.7×10^6	5×10^6	6.25×10^6
Data storage	22 MB	40 MB	50 MB
Model parameters	4 MB	4 MB	4 MB
OPS	108 BOPS	100 BOPS	125 BOPS

Table 3.3 Computing and memory associated with preliminary target detector stage versus pixel rate.

Third-Stage Target Detection/Classification: This stage processes only those pixels which have passed through the above two screening stages. As shown in the PFA/PFR curves in Figure 3.7, the preliminary detector further reduces the number of objects by another factor of approximately 25 at the PFR of about 0.3%. Thus this stage is required to process a total of only 2000 objects per second. Since this is only a fraction of the total data, fine resolution (best available), as well as sophisticated processing per pixel can be afforded. We have used approximately 1000 surrounding pixels per target and 100 features per target and full covariance GMMs with 60 components. Since we have assumed 30 different target types present in the data set, we need to evaluate 30 LLR decision metrics and 31 PDFs per object. Each decision metric requires $2 \times 100 \times 101 \times 60$ operations per object. Also, $2 \times 100 \times 1000$ operations are required to extract feature vectors per object. Thus a total of approximately 76 BOPS are to be performed at this stage.

Each object from preliminary detector is passed to this stage along with its surrounding pixels within a target size box of 1000 pixels. Thus this stage is required to store 8 MB of data. Also, approximately 75 MB of local memory per processor is required to store eigenvectors, 31x60 covariance matrices (each 100x100), mean vectors, and weights. Computing and memory requirement at this stage is again presented in the following Table 3.4 for three pixel rates.

Target/Clutter Classifier	2×10^6 pix/sec.	2×10^7 pix/sec.	2×10^8 pix/sec.
Resolution Used	3 x 3 sq.ft.	1 x 1 sq.ft.	0.5 x 0.5 sq.ft.
Surrounding pix/tgt	111	1000	4000
Objects to be processed	600	2000	3400
Pixels to be stored	7×10^4	2×10^6	1.36×10^7
Data storage	0.3 MB	8 MB	55 MB
Model parameters	75 MB	75 MB	77 MB
OPS	22 BOPS	76 BOPS	127 BOPS

Table 3.4 Computing and memory associated with target/clutter classifier stage versus pixel rate.

Final Target Hypothesis Deconfliction: This stage processes only those pixels which have passed through all of the above stages. The target versus background classifier even further reduces the number of objects by another factor of approximately 5. Only about 400 objects per second are passed to this stage for target versus target deconfliction. Since this is an even smaller fraction of the total data, an even more sophisticated level of processing can be afforded.

This stage is quite similar to the previous stage except that it classifies one target versus another target as opposed to any target versus clutter. In this stage also, we have used approximately 1000 surrounding pixels per target, 100 features per target and full

covariance GMMs with 60 components. If the indexes of the top 15 probable targets are provided from the previous stage, then a maximum of 105 possible target-target pairs per object of interest are required to be tested. To extract feature vectors $105 \times 2 \times 100 \times 1000$ operations per object are required. Also it requires approximately 103 BOPS to evaluate 105 LLR decision metrics for all 400 objects. This implements a logic to deconflict different target types where the first target competes with rest of all targets, then the first winner competes with remaining winners, and the steps are repeated until the final winner is found. Using this method, one ends up evaluating much less than 105 pairs per object on average. Thus this stage performs a maximum total of approximately 112 BOPS.

The previous stage passes about 400 objects along with their surrounding pixels within a target size box of 1000 pixels each. Thus this stage needs to store a minimal amount of data which is only 1.6 MB. On the other hand, it has maximum amount of model parameters of about 2.4 GB, which include mean vectors, covariance matrices, weights, and eigen vectors. Summary table 3.5 of this final stage is provided below for three pixel rates.

Target/Target Classifier	2×10^6 pix/sec.	2×10^7 pix/sec.	2×10^8 pix/sec.
Resolution used	3 x 3 sq.ft.	1 x 1 sq.ft.	0.5 x 0.5 sq.ft.
Surrounding pixs/target	111	1000	4000
Objects to be processed	400	400	640
Pixels to be stored	4.4×10^4	4×10^5	2.56×10^6
Data storage	0.2 MB	1.6 MB	10 MB
Model parameters	2.22 GB	2.4 GB	3 GB
OPS	104 BOPS	112 BOPS	220 BOPS

Table 3.5 Computing and memory for target/target stage versus pixel rate.

Total number of objects to be processed, required data storage space, statistical model parameters storage space and operations per second are summarized in table 3.6 for each stage of our end-to-end processing chain at a given pixel rate.

Input pixel rate: 2×10^7 pixels/sec
Input Resolution: 1x1 sq ft
Number of Target types: 30

Processing Stages	Anomaly Detector	Preliminary Target Detector	Target Detector & Preliminary Classifier	Target Classifier
Spatial Resolution	1x1 sq ft	4x4 sq ft	1x1 sq ft	1x1 sq ft
Surrounding pixels		100 pixels/target	1000 pixels/target	1000 pixels/target
Objects to be Processed	2.00E+07	5.00E+04	2000	400
Pixels to be Stored	2.00E+07	5.00E+06	2.00E+06	4.00E+05
Data Storage	80 MB	40 MB	8 MB	1.6 MB
Model Parameters	2 MB	4 MB	75 MB	2.4 GB
Operations per sec.	1.12 Gops	100 Gops	76 Gops	112 Gops

Table 3.6 Summary of computation power and memory for a pixel rate of 2×10^7 .

As mentioned before, we have also carried out the analyses for two other pixel rates and final numbers are also provided here in table 3.7 and 3.8.

Input pixel rate: 2×10^8 pixels/sec
Input Resolution: 0.5x0.5 sq ft (2 polarization)
Number of Target types: 30

Processing Stages	Anomaly Detector	Preliminary Target Detector	Target Detector & Preliminary Classifier	Target Classifier
Spatial Resolution	1x1 sq ft	4x4 sq ft	0.5x0.5 sq ft	0.5x0.5 sq ft
Surrounding pixels		100 pixels/target	4000 pixels/target	4000 pixels/target
Objects to be Processed	5.00E+07	6.25E+04	3400	640
Pixels to be Stored	2.00E+08	6.25E+06	1.36E+07	2.56E+06
Data Storage	800 MB	50 MB	55 MB	10 MB
Model Parameters	2 MB	4 MB	77 MB	3.0 GB
Operations per sec.	2.8 Gops	125 Gops	127 Gops	220 Gops

Table 3.7 Summary of computation power and memory for a pixel rate of 2×10^8 .

Input pixel rate: 2×10^6 pixels/sec
Input Resolution: 3x3 sq ft
Number of Target types: 30

Processing Stages	Anomaly Detector	Preliminary Target Detector	Target Detector & Preliminary Classifier	Target Classifier
Spatial Resolution	3x3 sq ft	3x3 sq ft	3x3 sq ft	3x3 sq ft
Surrounding pixels		178 pixels/target	111 pixels/target	111 pixels/target
Objects to be Processed	2.00E+06	1.50E+04	600	400
Pixels to be Stored	2.00E+06	2.67E+06	7.00E+04	4.40E+04
Data Storage	8 MB	22 MB	0.3 MB	0.2 MB
Model Parameters	2 MB	4 MB	75 MB	2.22 GB
Operations per sec.	112 Mops	108 Gops	22 Gops	104 Gops

Table 3.8 Summary of computation power and memory for a pixel rate of 2×10^6 .

3.2.4 Real time Implementation:

The methods described above require appreciable computing as well as large amount of memory to store image data and/or model parameters. This requires implementations in distributed multi-processor architectures, with attention to the mapping of the processing, memory, and interprocessor communications in a manner consistent with available hardware technologies. For this we have mainly studied the case with a total throughput of 2×10^7 pixels per second.

We have reviewed available hardware technologies to design a preliminary architecture of ATR processing. The current 32-bit wide PCI Bus offers a bandwidth of 132 MB/sec at 33 MHz. The 66 MHz 64-bit wide version is expected to be available in the very near future, providing up to 533 MB/sec. The VME64 Bus offers a bandwidth of 80 MB/sec, while two-edge 2eVME64 has reached data rates of 160 MB/sec. Existing VME320 offers a bandwidth of 320 MB/sec and is projected to be capable of operating at as high as 533 MB/sec without errors. Estimated speeds of future generations of VME Bus are more than 1 GB/sec by the year 2000. On the above basis, we have made a conservative assumption for the bus speed of 400 MB/sec which is expected to be well within the range of both fast-wide PCI and also the next generation of VME beyond VME320.

We have also reviewed available fixed-point and floating-point digital signal processors (DSPs). The current generation of Texas Instrument's C67xx series offers 1 GFLOPS and TI envisions its next generation to be 3 GFLOPS by year 2000. Quad C6701 cards currently offer 4 GFLOPS using a single-slot 6U VME board. The SHARC processor from Analog Device offers 120 MFLOPS, and a single 6U VME board with 24 SHARCs providing up to 2.88 GFLOPS is also available. Based on the above, we have assumed a sustained throughput of each processing unit to be 2 GFLOPS.

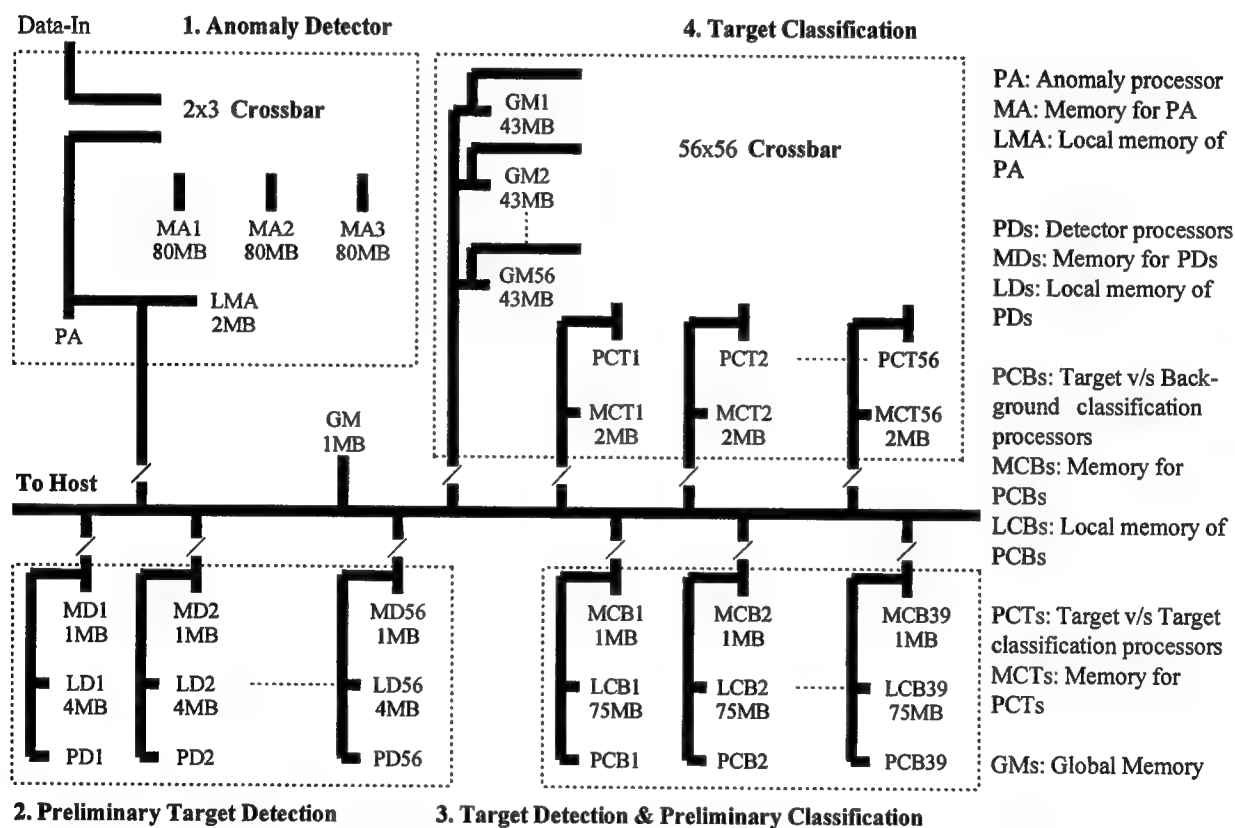


Figure 3.12 Sample distributed architecture for implementing the end-to-end ATR.

We have designed a preliminary distributed processing architecture for implementing the end-to-end ATR processing using the present approach and including E/O interconnect crossbar technologies as illustrated in Figure 3.12. The main roles of the different processing units are indicated in the figure (Anomaly Processor [PA], Second Stage

Detection Processors [PD], Third Stage Target vs. Background Classification Processors [PCB], and Final Stage Target Hypothesis Deconfliction Processors [PCT]). The memories associated with these processors are also shown.

The first stage of anomaly detector (1.12 GOPS) requires only one processor [PA] and can finish the anomaly detection job in about 560 ms. Only 100 ms is required to transfer 40 MB of input data to the next stage at 400 MB/S bus speed. Since PA is idle for more than 300 ms, it performs the subsampling (4x4 sq.ft.) on input raw data and passes subsampled data to detector stage. When the detector stage finishes target detection, it writes the LLR metrics for each object into global memory (GM), which then are read and sorted by PA and probable target-like objects with their surrounds (8 MB) are passed onto the third stage with original resolution. Total time required for all of the back and forth data transfer is about 125 ms.

Since the anomaly detector is required to send coarse data of the current frame to the detector stage and fine resolution data of the previous frame to target/clutter classifier stage, it has to have an access to current frame data and one frame of prior data. Because of this, we have provided three memory banks (each 80 MB) in the anomaly detector stage. As shown in above figure 3.12, a small 2x3 crossbar switch offers an efficient solution to continuously feed the data at this first stage. While the current data is being written to bank 1, PA is working on data in bank 2 and PA can still have an access to even prior data in bank 3 when they are required to be passed onto the third stage. The anomaly detector stage is very general, and scene and target type independent. It does not require any kind of target model parameters to be loaded in its local memory. It only needs less than 2 MB to store its own compiled source code.

The second stage of preliminary target detector has to wait 100 ms before it gets data from the anomaly detector. Once it finishes preliminary target detection, it writes three LLR metrics for all anomalous pixels (0.6 MB) to global memory. Thus this stage has to

perform about 100 GOP in approximately a little less than 900ms. We need at least 56 processor units [PD] for this stage. Total local memory to store data is 40 MB which results in less than 1 MB per processor. Also we need 4 MB per processor to store model parameters and compiled source code of this stage. Thus a total of 280 MB memory is required in this stage.

Here we have two choices to implement a real time architecture. One is to provide a crossbar interconnection between processors and memory elements where model parameters are stored. Since we have an optimal number of processors, we can not afford any time for model parameter transfers. So we would need a 56x56 crossbar switch. But all we can save is 110 MB of memory which is required to provide an individual copy of model parameters to each processor. The alternative, a less expensive and equally efficient approach, is to provide an individual copy of model parameters to each processor and not to use crossbar switching here. The latter approach has been assumed in Figure 3.12. A similar decision will be made for the next stage, as described below.

The third stage of target versus clutter classifier receives data only after the second stage computes LLRs for all anomalous pixels. Once the decision metrics from the detector stage are in global memory, PA sorts them and passes about the top 2000 objects detected by the second stage along with their surround to this third stage. This data transfer of 8 MB takes only about 20 ms. At the end of the task, detected target locations along with the index of top 15 most probable targets (less than 2 MB) are passed on to the target-target classifier, which takes about 4 ms. Thus this stage has to perform about 76 GOP in approximately 975ms. We need at least 39 processor units [PCB] for this stage. This stage requires 75 MB to store GMM parameters and eigenvectors. Also 8 MB is needed to store all data i.e. less than a MB per processor. Thus a total of approximately 3 GB memory is required at this stage.

Again we have same two options available for real time implementation as in the previous stage. One is to implement a 39x39 crossbar switch while the other is to provide an individual copy of model parameters to each processor. If we use a crossbar we can only save up to 2.85 GB of memory which may still be less expensive than a 39x39 crossbar. As shown in figure 3.12, we again elected to provide an individual copy to each processor rather than using a crossbar configuration at this stage.

Final target versus target classification stage receives data (less than 2 MB) in less than 5 ms. This stage demands very high computing as well as very high storage. It has to performs about 112 GOPS in little less than a second. We need at least 56 processor units [PCT] for this stage. Memory required to store data is minimal (a total of less than 2 MB), but we need 2.4 GB per processor to store model parameters for all 435 possible target-target pairs.

At this stage, if we do not use a crossbar configuration, then the individual copy of model parameters to each processor costs about 135 GB (more than \$ 135,000). On the other hand, if we use a crossbar configuration of 56x56, we can have just one copy of model parameters distributed in 56 different memory modules with each module of about 43 MB. Also, it is to be noticed that the logic to deconflict target-target hypothesis described in the previous subsection (3.2.4) does not require all 105 possible pairs to be evaluated. According to our analysis, this reduction in computing should compensate for time consumed in model data communication. As shown in the above architecture in Figure 3.12, a 56x56 crossbar switch would appear to be an efficient solution.

4. DISCUSSION

With regard to the high resolution and wide area search SAR image formation applications discussed in Section 2, the case for high speed interconnects is very clear. Future availability of a non-blocking crossbar switch with on the order of 400 MB/s or more throughput per channel, and 10-30 "input" versus 20-140 "output" nodes has potential to allow these SAR imaging applications to be executed with on the order of 20% or less overhead in terms of the total amount of raw computing required.

To some degree this was the expected conclusion, but here it has been verified by analyses of specific examples. Of special note is that, even with very high speed processing nodes, the required crossbar switch size is also rather large. It would be even larger with less capable processing nodes. Conversely, as processing node performances continue to increase the required crossbar switching size may become less for the same applications, while the bandwidth of each switched channel will be required to increase.

For the basic SAR imaging computing structures addressed above, the envisioned crossbar configurations tend to be highly asymmetric. This would not allow for efficient utilization of the resources of the types of fully symmetric E/O crossbar switches which we understand to be currently under development in the DARPA VLSI Photonics program. It is expected, however, that methods for more efficiently mapping the same processing onto a symmetric switch can be devised. For SAR systems with multiple polarization channels, a fully symmetric mapping of the processing for two parallel channels onto a single switch is also possible. At some additional cost it would also be possible to use multiple switches to separately deal with multiple overlapping synthetic apertures on the same polarization channel, the advantage again being that the switches could then

become more symmetric (and also smaller). The benefits in switch cost might well overcome the penalties in additional processing overhead.

As a final note on the SAR processing studies, as previously explained, the specific image formation algorithm addressed here was a stripmap formulation, which actually requires less in the way of large-scale data rearrangements (for a given sub-image size) than the competing spotlight approaches reviewed in Section 2, which also tend to be more limited in their capabilities to support wide area search. Future implementations of these algorithms, or improved methods derived therefrom, could potentially benefit even more from the types of high speed crossbar switching discussed in the present report.

The apparent benefits of high speed crossbar switching for the types of ATR algorithms and applications discussed in Section 3 are more complex, due in part to the greater complexity of the algorithms themselves. In addition, and mainly in response to this algorithm complexity, specific processing architectures presented in Section 3 were highly “pipelined” to accommodate specific algorithm and application details. Such highly pipelined computing architectures, if realizable, tend to minimize the need for more general and flexible crossbar switching for large data rearrangements.

Even so, in the main ATR case discussed above, it has been shown that crossbar switching allows very efficient processor utilization and also saves more than 100 GB of memory. We have also separately estimated the computing, memory and bandwidth requirements using a pixel rate of 2×10^8 per second. That study indicates that such future generations of ATR processing may benefit even more from E/O interconnect technology. In addition, since the above hypothesized computing architecture is highly specialized and “pipelined” to the specific application, and even to the number of target hypotheses to be considered, it

actually minimizes the need for switched general purpose interconnects. In a more general architecture, and in providing flexibility for multiple different algorithm and application details, the need for high speed E/O crossbar switching can be expected to be even greater than indicated by the above.

A very similar architecture to the above could be envisioned for implementation of major aspects of the MSTAR ATR approach. Minor differences would be encountered in the first three stages. A major difference would be in the final stage, where the MSTAR approach uses 3D target models stored in memory and generates templates from these target models on the fly. Therefore, the distributed model memory in the fourth stage of the above would likely be replaced by a replicated memory of the 3D target models (which are fairly compact), but now also including an added processing unit for generating the target templates. These target model memories and their processing units would be replicated as many times as needed to meet the problem throughput, and would be crossbar-connected to the PCT processors much as already shown above.

Another observation is that the crossbar configuration for the ATR architecture outlined above involves communications from processing units on one side to memory units on the other. In general this requires that the network interface units between the crossbar and the rest of the system support such capability, rather than simply supporting processor-to-processor communications. It is not clear whether all of the approaches currently under development in the VLSI Photonics program envision providing this capability, rather than just inter-processor communication by MPI (Message Passing Interface) standards. At this time we understand that Northrop envisions processor-to-memory interfacing, while Honeywell does not. In an MPI based approach, the architecture in Figure 3.12 would need to be modified to collocate the processors with the distributed memory, and would

likely be less efficient as a result. This, however, would not seem to be the case for implementing the MSTAR ATR approach as discussed above.

As a final point, while the ATR architecture outlined above depicts two crossbars of different sizes, we envision that their functions would be done in practice by a single, larger unit. Then, with a highly pipelined architecture, there would be redundancy of the ultimate interconnect capabilities relative to what is required. This suggests that there may be a useful tradeoff between the tolerable fraction of unusable crossbar interconnections due to manufacturing difficulties versus the degree to which the actual computing architecture can be optimally mapped onto the "good" paths of the manufactured switch.

DISTRIBUTION LIST

addresses	number of copies
AIR FORCE RESEARCH LABORATORY/SNDP CPT KEVIN WHITCOMB 25 ELECTRONIC PKY ROME NY 13441-4515	5
DEFENSE GROUP INC 19C CROSBY DRIVE STE 350 BEDFORD MA 01730	5
AFRL/IFOIL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-OCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	2
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ATTN: NAN PFRIMMER IIT RESEARCH INSTITUTE 201 MILL ST. ROME, NY 13440	1
AFIT ACADEMIC LIBRARY AFIT/LDR, 2950 P. STREET AREA B, BLDG 642 WRIGHT-PATTERSON AFB OH 45433-7765	1
AFRL/MLME 2977 P STREET, STE 6 WRIGHT-PATTERSON AFB OH 45433-7739	1

ATTN: SMDC IM PL 1
US ARMY SPACE & MISSILE DEF CMD
P.O. BOX 1500
HUNTSVILLE AL 35807-3801

COMMANDER, CODE 4TL000D 1
TECHNICAL LIBRARY, NAWC-WD
1 ADMINISTRATION CIRCLE
CHINA LAKE CA 93555-6100

CDR, US ARMY AVIATION & MISSILE CMD 2
REDSTONE SCIENTIFIC INFORMATION CTR
ATTN: AMSAM-RD-DB-R, (DOCUMENTS)
REDSTONE ARSENAL AL 35898-5000

REPORT LIBRARY 1
MS P364
LOS ALAMOS NATIONAL LABORATORY
LOS ALAMOS NM 87545

AFIWC/MSY 1
102 HALL BLVD, STE 315
SAN ANTONIO TX 78243-7016

USAF/AIR FORCE RESEARCH LABORATORY 1
AFRL/VSOSA(LIBRARY-BLDG 1103)
5 WRIGHT DRIVE
HANSCOM AFB MA 01731-3004

ATTN: EILEEN LADUKE/D460 1
MITRE CORPORATION
202 BURLINGTON RD
BEDFORD MA 01730

OUSD(P)/DTSA/OUTD 1
ATTN: PATRICK G. SULLIVAN, JR.
400 ARMY NAVY DRIVE
SUITE 300
ARLINGTON VA 22202

RICHARD PAYNE 1
AIR FORCE RESEARCH LAB/SNH
HANSCOM AFB, MA 01731-5000

JOSEPH P. LORENZO, JR.
AIR FORCE RESEARCH LAB/SNHC
HANSCOM AFB, MA 01731-5000

1

JOSEPH L. HORNER
AIR FORCE RESEARCH LAB/SNHC
HANSCOM AFB, MA 01731-5000

1

RICHARD A. SOREF
AIR FORCE RESEARCH LAB/SNHC
HANSCOM AFB, MA 01731-5000

1

ALBERT A. JAMBERDINO
AIR FORCE RESEARCH LAB/IFED
32 HANGAR RD
ROME NY 13441-4114

1

AIR FORCE RESEARCH LAB/SND
25 ELECTRONIC PKY
ROME NY 13441-4515

1

JOANNE L. ROSSI
AIR FORCE RESEARCH LAB/SNW
25 ELECTRONIC PKY
ROME NY 13441-4515

1

NY PHOTONIC DEVELOPMENT CORP
MVCC ROME CAMPUS
UPPER FLOYD AVE
ROME, NY 13440

1

ROBERT T. KEMERLEY
AIR FORCE RESEARCH LABORATORY/SND
2241 AVIONICS CIRCLE, RM C2G69
WRIGHT-PATTERSON AFB OH 45433-7322

1

***MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)***

The advancement and application of information systems science and technology for aerospace command and control and its transition to air, space, and ground systems to meet customer needs in the areas of Global Awareness, Dynamic Planning and Execution, and Global Information Exchange is the focus of this AFRL organization. The directorate's areas of investigation include a broad spectrum of information and fusion, communication, collaborative environment and modeling and simulation, defensive information warfare, and intelligent information systems technologies.